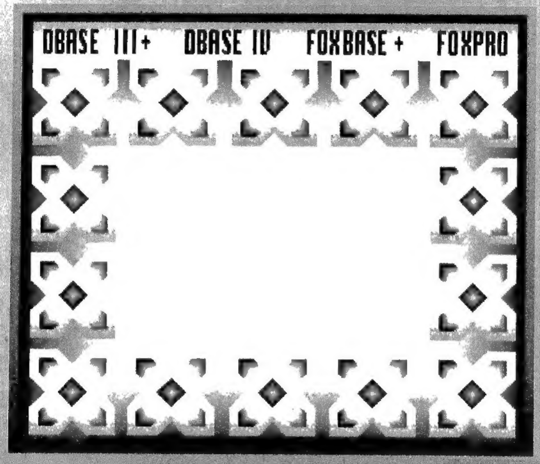
نظم إدارة قواعد ال

الجزء الأول



م. مصطفى رضا عبد الوهاب — أ.د. محمد فهمى طلبه

علاء الدين محمد فهمى — أ.د. عمرو جنيد

عبد العزيز حسن الحريرى — أ.د. محمد على الشرقاوى



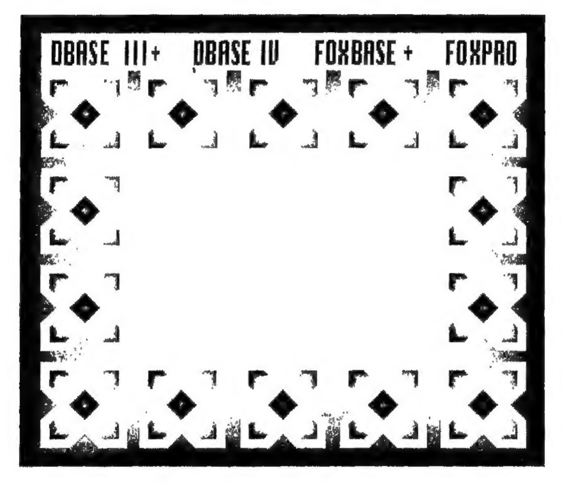
٥

موسوعة دلتا كمبيوتر


نظم إدارة قواعد البيانات

نظم إدارة قواعد البيانات

الجزء الأول



م . مصطفى رضا عبد الوهاب	ا .د. محمد فهمى طلبه
د . علاء الدين محمد فهمى	ا.د. عمرو جنيد
م . عبد العزيز حسن الحريرى	ا.د. محمد على الشرقاوى

٥

موسوعة دلتا كمبيوتر

تقـــــديم

ان التطور في لغات البرمجة للحاسب الالكتروني قد مر بمراحل متعددة أرتبطت بعدة عوامل من أهمها التطور التكنولوجي المتلاحق في مكونات الحاسب المادية وما تبع ذلك من زيادة سرعة عمليات الحاسب المختلفة وزيـــادة كفاءتها . وقد أدى هذا الى تطور هائل فى نظم التشغيل وظهور مفاهيم جديدة مثل تعدد الوظائف وتعدد المستخدمين واستخدام الشبكات بالاضافة الى العديد من الخصائص التى تميز الاجيال المتلاحقة من الحاسب الالكتروني .

ومن الملامح الرئيسية لتطور لغات البرمجة البعد التدريجي عن التدخل فى العديد من العمليات التفصيلية التى تتم بواسطة الحاسب لأداء مهمة معينة مما أتاح الفرصة أمام مخططى البرامج لاستخدام خصائص الحاسب دون الحاجة الى الالمام الكامل بعملياته الداخلية . ومن المعروف أن الحاسب حقيقة لا يفهم الا لغة الواحد والصفر وهى ما يطلق عليها لغة الماكينة (Machine Language) . وكان مخطط البرامج لا يستطيع التفاهم مع الحاسب الا من خلال هذه اللغة المعقدة . ثم ظهرت لغات أكثر سهولة قامت بتكوين مجموعات من الواحد والصفر في رموز بسيطة واستخدام هذه الرموز فى كتابة البرامج . وتسمى هذه اللغات باللغات الرمزية (Symbolic Languages) أو لغات التجميع (Assembly Languages) . وأخذت اللغات فى التطور مع زيادة درجة التمثيل حتى وصلت الآن الى لغات الجيل الرابع(4th Generation Languages) حيث أتاحت هذه اللغات لمخطط البرامج التعامل مع النوافذ والقوائم الواضحة التى يستطيع من خلالها تصميم البرامج المطلوبة بسهولة تامة ويسر .

وقد انعكس هذا التطور الكبير على المتخصصين فى مجال الحاسب حيث أصبح على مخططى البرامج متابعة كل جديد فى مجال نظم تطوير البرامج وأدواتها المتقدمة وذلك حتى يمكنهم الاستفادة من خصائصها في تصميم النظم المتميزة التى توفر الكفاءة العالية وسهولة الاستخدام . ومن ناحية أخرى فقد أصبح على المستخدم ضرورة الالمام بهذه البرامج التطبيقية الحديثة حتى يستطيع الاعتماد على نفسه فى الاستفادة منها والاضافة اليها . وقد أدى ذلك الى ظهور جيل جديد من المستخدمين الذين يمتلكون خبرة كبيرة في التعامل مع العديد من البرامج التطبيقية الى جانب القدرة على تصميم النظم الخاصة بهم .

وقد كان لهذا التطور فى مجال الحاسبات أثره في طبيعة الكتب المتخصصة على المستوى العالمى فى هذا المجال . فبعد أن كانت هذه الكتب - الى وقت قليل مضى - تركز على الجوانب النظرية ، أصبحت الآن تركز على أساليب استخدام التطبيقات وعلى تقديم الخبرات والمهارات العملية لمستخدمى الحاسبات .

وقيام مؤسسة "دلتا" بتقديم موسوعتها الجديدة تكنولوجيا وعلوم الحاسب يعوض النقص الشديد الذى تعانى منه المكتبة العربية فى هذا المجال حيث أن معظم الكتب العربية الموجودة ليست سوى ترجمة أو تلخيص سطحى لدليل التشغيل لنظم الحاسب المختلفة بينما يحتاج المستخدم الى توضيح الكثير من الجوانب العلمية والفنية

بالاضافة الى خصائص تشغيل النظم . وهذا الجهد المشكور من مؤسسة دلتا هو امتداد لسياستها الواعية واحساسها بمستوليتها نحو التطور التكنولوجي بالمنطقة العربية .

وهذا الكتاب هو أحد كتب موسوعة "دلتا" لتكنولوجيا و علوم الحاسب . وهو يمثل حلقة الاتصال بين الجوانب التطبيقية والجوانب العلمية والفنية . فبالرغم من تركيزه على شرح خصائص تشغيل نظم عائلة (DBase) ، الا أنه اهتم بتوضيح مفهوم قواعد البيانات بصفة عامة وتوضيح العمليات المرتبطة بها مثل الفهرسة والبحث عن البيانات وأساسيات تصميم البرامج ، و ..... الخ .

ومما لاشك فيه أن موضوع هذا الكتاب يعد من أهم الموضوعات التي تشغل أذهان جميع المتخصصين في مجال الحاسب وذلك لارتباطه المباشر بالمجالات العملية سواء للأفراد أو للمؤسسات . فقد أصبحت قواعد البيانات وتطبيقاتها تغطي معظم مجالات الحياة وأصبح استخدام الحاسب في ادارة البيانات والسيطرة عليها أمرا مألوفا في كل موقع . ورغم توفر عدد كبير من البرامج التطبيقية التي تخصصت في ادارة قواعد البيانات ، مثل (Informex) ، (Oracle) ، (Paradox) ، (Focus) ، .... الخ الا أن مؤسسة "دلتا" اختارت برامج عائلة (DBase) لتكون موضوع هذا الكتاب بأجزائه الثلاث وذلك لما تتمتع به عائلة (DBase) من الشيوع والانتشار وسهولة الاستخدام بالاضافة الى قدرتها على التعامل مع العديد من نظم التشغيل .

أ. د. محمد فهمي طلبه

محتــويات الكتــاب

مسلسل	الموضوع	رقم الصفحة

مسلسل	الموضوع	رقم الصفحة

مسلسل الموضوع رقم الصفحة

مسلسل	الموضوع	رقم الصفحة

مسلسل	الموضوع	رقم الصفحة
٢٦ - ٢	أوامر التصحيح (Debugging Commands)	٣١٨
٢٦ - ٣	تعليق تنفيذ البرنامج (Suspend)	٣١٨
٢٦ - ٤	استخدام مخزن التاريخ (History)	٣١٩
٢٦ - ٥	مراقبة تنفيذ البرنامج	٣٢٠
٢٦ - ٦	الأمر (SET TALK ON)	٣٢٠
٢٦ - ٧	الأمر (SET ECHO ON)	٣٢٠
٢٦ - ٨	الأمر (SET STEP ON)	٣٢٠
٢٦ - ٩	الأمر(SET DEBUG ON)	٣٢٠
٢٦ -١٠	عرض محتويات الذاكرة (Display Memory)	٣٢١
٢٦ -١١	عرض الحالة (Display Status)	٣٢١
٢٦ -١٢	عرض تركيب ملف قاعدة البيانات	٣٢١
	الملاحق	٣٢٣
	ملحق (١) أهم الأوامر والدوال المستخدمة فى برنامج (DBase IV)	٣٢٥
	أولا : الأوامر	٣٢٧
	ثانيا : أوامر التجهيز (Set)	٣٣٢
	ثالثا : الدوال	٣٣٤
	ملحق (٢) أهم الأوامر والدوال المستخدمة فى برنامج (+ FoxBase)	٣٣٧
	أولا : الأوامر	٣٣٩
	ثانيا : الدوال	٣٤١
	ملحق (٣) أهم الأوامر والدوال المستخدمة فى برنامج (FoxPro)	٣٤٥
	أولا : الأوامر	٣٤٧
	ثانيا : الدوال	٣٤٩
	ملحق (٤) جدول مقارنة نظم ادارة قواعد البيانات	٣٥١

الباب الأول

مقدمة عامة

١ - ١ هذا الكتاب

هذا الكتاب هو الجزء الأول من " نظم ادارة قواعد البيانات " الذى يتكون فى الواقع من ثلاثة أجزاء . وفى هذا الجزء احاطة كاملة ببرامج عائلة (DBase) التى تشمل البرامج (+ DBase III ، DBase IV ، FoxBase + ، FoxPro) . وهو يتضمن شرح تفصيلى دقيق لهذه البرامج وطريقة تشغيلها والقوائم المستخدمة فيها . وقــد روعى فى الشرح توضيح خطوات كل عملية يتم اجراؤها بالتسلسل المنطقى الدقيق .

كما يتضمن الكتاب أيضا شرح مبادىء البرمجة بصفة عامة ثم تطبيق هذه المبادىء على كتابة البرامج بواسطة برامج عائلة (DBase) . بالاضافة الى شرح شامل لمكونات البرامج متضمنا الأمثلة الواضحة وشرحها التفصيلى .

ورغم أن هذا الجزء يعتبر كافيا للاحاطة الكاملة ببرامج عائلة (DBase) ، الا أن الجزء الثانى يعتبر ضروريا للقارىء الذى يريد اكتساب مهارة عالية والانتقال الى مرحلة الاحتراف حيث أنه يتضمن جميع الأوامر والدوال المستخدمة والشرح التفصيلى لها . أى أنه يعتبر مرجعا شاملا يمكن الرجوع اليه فى أى وقت للتعرف على الشكل السليم (Syntax) لأى أمر .

أما الجزء الثالث الذى ينشر تحت عنوان " تطبيقات نظم ادارة قواعد البيانات " فيعتبر مرحلة متقدمة جدا تفيد مخططى البرامج الذين يمتلكون خبرة كبيرة بنظم أدارة قواعد البيانات ، ويريدون تصميم نظم كاملة تخدم أى موقع .

وهذا الكتاب يتكون من ستة وعشرين بابا يتم من خلالها شرح قوائم المساعد (Assistant) المستخدمة فى برنامج (+ DBase III) وطريقة تشغيل البرنامج من الألف الى الياء . ثم ينتقل الكتاب الى شرح كيفية كتابة البرامج من خلال مشيرة النقطة (Dot Prompt) مع شرح كافة الأوامر والدوال المستخدمة .

كما يتضمن الملحق الموجود فى آخر الكتاب ملخصا لجميع الأوامر والدوال المستخدمة فى برامج (DBase IV) و (+ FoxBase) و (FoxPro) .

والكتاب الثانى ، كما سبق الايضاح ، هو جزء مكمل لهذا الكتاب لانه يشمل الشرح الدقيق لجميع الأوامر والدوال المستخدمة فى برنامج (+ DBase III) وكذلك الأوامــر والــدوال المستخــدمة فــى برامج (DBase IV) ، (+ FoxBase) ، (FoxPro) .

لذلك يستطيع القارىء اقتناء الجزئين الأول والثانى معا حتى يضمن الالمام الكامل ببرامج عائلة (DBase) . كما يمكنه اقتناء الكتاب الأول كبداية . ثم شراء الكتاب الثانى عندما يحتاج اليه .

أما الكتاب الثالث فلا ينصح بشرائه الا للقارىء الذى يرغب فى الوصول الى درجة

كبيرة من المهارة فى التعامل مع برامج عائلة (DBase) والقدره على تصميم النظم الكاملة .

١ - ٢ ما هى قاعدة البيانات ؟

رغم أن اسم قاعدة البيانات غير مألوف للانسان العادى البعيد عن مجال الحاسب ، الا ان كل انسان يقابل قواعد البيانات يوميا . فعندما يبحث الانسان فى دليل التليفون مثلا فانه يتعامل مع قاعدة بيانات . وعندما يسجل بيانات خاصة بالمعارف والأصدقاء فى نوتة معينة للرجوع اليها عند الحاجة فانه ينشىء قاعدة بيانات. وعندما يبحث موظف معين عن بيانات أحد العملاء عن طريق الدوسيهات الموجودة لديه فانه يتعامل مع قاعدة بيانات . وهكذا فان تعامل الانسان مع قواعد البيانات يأخذ صورا وأشكالا متعددة لايمكن حصرها .

وعندما يقوم موظف الأرشيف بتنظيم البيانات الموجودة لديه . فانه يخصص دوسيه مثلا أو دفترا لكل مجموعة من البيانات التي تخص موضوعا معينا مثل دفتر الحضور والانصراف ، ودفتر البيانات الشخصية ، ودفتر الشئون المالية ،... الخ . كمايقوم الموظف أيضا بتخصيص صفحة فى الدوسيه أو الدفتر لكل موظف بالشركة . وفى صفحة الموظف يقوم بتخصيص أعمدة تمثل بيانات تاريخ الحضور وتاريخ الانصراف و... الخ .

وما يحدث مع الحاسب لايختلف كثيرا عن ذلك .. حيث يتم انشاء ملف خاص لكل مجموعة من البيانات التى تخص موضوعا معينا وهو يقابل الدفتر الذى ينشؤه الموظف . ثم يتم انشاء سجل خاص بكل موظف وهو يقابل الصفحة التى يتم تخصيصها لكل موظف فى الدفتر أو الدوسيه . ثم يتم تسجيل بيانات كل موظف فى السجل الخاص به فى حقل البيانات الذى يمثل كل بيان مطلوب ادخاله .

ولكن هناك اختلافا واضحا بين انشاء قاعدة البيانات بواسطة الانسان وانشائها بواسطة الحاسب . حيث أن الانسان مثلا يمكنه مباشرة التمييز بين البيانات الموجودة داخل السجل .

فمثلا بالنسبة للانسان يكون واضحا أن (Mohamed) تمثل اسما وليس رقم تليفون . فى حين لايستطيع الحاسب تمييز ذلك الا عن طريق وضع قواعد معينة تجعله يستطيع التمييز بين البيانات الحرفية والبيانات العددية مثلا. لذلك فان انشاء قاعدة البيانات للحاسب يجب أن يخضع لقواعد معينة . كما يجب أن تتميز هذه القواعد بالوضوح الشديد . وذلك لأن الحاسب رغم سرعته الفائقة فى تنفيذ العمليات ، الا أنه لايتمتع بأى قدر من الذكاء .

لذلك فان تصميم هيكل قاعدة البيانات (Structure) يبدأ بتحديد المعلومات المطلوب تخزينها . ثم يتم تقسيم هذه المعلومات الى وحدات بيانات صغيرة مثل الاسم والعنوان ورقم التليفون و... الخ . وحيث أن هذه الوحدات تكون مشتركة فى جميع السجلات ، لذلك يتم تعريفها للحاسب كحقول . كما يتم تعريف الحاسب بنوع البيانات الموجودة داخل هذه الحقول .

١ - ٣ ماهى ادارة قواعد البيانات ؟

عندما يبحث الانسان عن بيان معين داخل قاعدة البيانات أو عندما يقوم بترتيب الأوراق الخاصة بالموظفين داخل الدفاتر حتى يمكنه بسهولة الوصول الى أى موظف ، فانه يمارس عملا من أعمال ادارة قواعد البيانات . وبالنسبة للحاسب يحدث نفس الشيء حيث يتم وضع برنامج معين للحاسب ليقوم بادارة قاعدة البيانات المخزنة به . هذا البرنامج عادة يؤدى المهام التالية :

١ - اضافة بيانات جديدة لقاعدة البيانات .
٢ - ترتيب البيانات بترتيب معين .
٣ - البحث عن بيانات معينة داخل قاعدة البيانات .
٤ - تعديل البيانات .
٥ - عرض وطباعة التقارير.
٦ - مسح البيانات .

وهناك عدة برامج تم تصميمها لادارة قواعد البيانات أهمها وأكثرها شيوعا وانتشارا هو ما يمكن أن نسميه عائلة (DBase) . وهي العائلة التي بدأت ببرنامج (DBase II) ثم (DBase III) ثم (DBase III+) ثم (DBase IV) . ثم تلي ذلك ظهور برامج (FoxBase) و (FoxBase +) و (FoxPro) وهي تشتمل على نفس الأوامر والخصائص الفنية الخاصة بعائلة (DBase) كما تضيف اليها أوامر وخصائص جديدة تجعلها اكثر قوة وسرعة .

كما ظهرت عائلات أخري مثل (Oracle) ، (SQL) ، (Informex) ، (4TH Dimension) . وهذا الكتاب يتناول شرح خصائص أحد هذه البرامج وهو برنامج (DBase III+) و كذلك النسخه الحديثة منه (DBase IV) وذلك لشعبيته وشيوعه . كما تم اضافة ملحق فى آخر الكتاب يوضح أوامر برنامج (DBase IV) وبرنامج (FoxBase +) وبرنامج (FoxPro) . وسوف يلاحظ القارىء أن هذه الأوامر هي نفس أوامر برنامج (DBase III+) مضافا اليها مجموعة أخرى من الأوامر.

١ - ٤ برنامج (DBase III+)

هذا البرنامج يمثل النسخة الثالثة من برنامج (DBase) المستخدم مع الحاسبات الصغيره (Microcomputers) ، حيث سبقته برامج (DBase II) ، (DBase III) . وهو يمتاز عن النسخ السابقة بتوفير التفاعل والحوار بين المستخدم والحاسب من خلال القوائم . وهذه القوائم تظهر من خلال برنامج المساعد (Assistant) . هذا البرنامج يتيح للمستخدم تنفيذ العمليات المختلفة علي قاعدة البيانات دون الحاجة الى استعمال الأوامر (Commands) التى تتطلب قدراً كبيرا من الالمام بالبرنامج .

كما أن برنامج (DBase III+) يمتاز أيضا باضافة مجموعة كبيرة من الأوامر (Commands) ، والدوال (Functions) ، وأدوات التصحيح (Debugging Tools) التى توفر لمخطط البرامج المزيد من المرونة فى وضع البرامج الكبيرة .

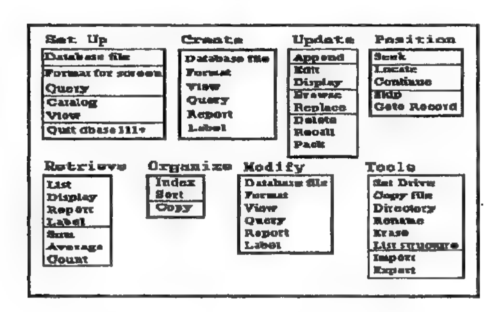
وبرنامج (DBase III+) ينقسم الي جزئين رئيسيين وهما برنامج المساعد (Assistant) ومشيرة النقطة (Dot Prompt) . وبرنامج المساعد يعتمد على القوائم التي تظهر علي الشاشة ويقوم المستخدم باختيار العملية المطلوب تنفيذها. أما مشيرة النقطة (Dot Prompt) فتتطلب أن يكتب المستخدم الأمر المطلوب تنفيذه والضغط على مفتاح الادخال .

ويلاحظ أنه عند استخدام برنامج المساعد يتم ظهور الأمر الذي يمثل العملية المنفذة عند مشيرة النقطة . وهذا يتيح للمستخدم التعرف على شكل الأمر (Syntax) قبل تنفيذه . وعادة يبدأ أي مستخدم لبرنامج (DBase III+) باستخدام برنامج المساعد. وعندما يكتسب الخبرة الكافية ينتقل الى استخدام مشيرة النقطة (Dot Prompt) الذي يقوده في النهاية الى كتابة البرامج الكبيرة لادارة قاعدة البيانات .

البابـ الثانى

انشاء ملف قاعدة البيانات

لتشغيل برنامج (DBase III+) يتم وضع قرص البرنامج في وحدة الأقراص الأولى ثم كتابة (DBase) وعند بدء تشغيل البرنامج تظهر الشاشة المبينة بالشكل (٢ - ١) . وهذه الشاشة تمثل قوائم برنامج المساعد (Assistant) .



شكل (٢ - ١) قوائم برنامج المساعد (Assistant)

ويلاحظ في هذا الشكل وجود مجموعة من القوائم (Menus) التي تظهر أسماؤها على الشاشة أعلى كل قائمة . وهذا الشكل يوضح القوائم وهي مفتوحة كلها وهذا للتوضيح فقط . لأن ما يحدث في الواقع أن القائمة التي يتم وضع المؤشر (Cursor) على اسمها فقط هي التي يتم فتحها أما باقي القوائم فتظل مغلقة . والقائمة التي يتم فتحها يظهر فيها مؤشر يمكن تحريكه حتى يصل الى الاختيار المطلوب والضغط على مفتاح الادخال ، فيتم تنفيذ هذا الاختيار .

٢ - ١ فتح القائمة

قبل اختيار أي أمر من القوائم يلزم أولا اختيار القائمة التي تحتوي على هذا الأمر . ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١ - يلاحظ في البداية وقوف المؤشر الخاص بعمود الاختيارات (Menu Bar) على القائمة الأولى (Set Up) . وبالتالي تظهر قائمة التجهيز (Set Up) أسفل العنوان وتحتوي على عدة اختيارات كما هو واضح من الشكل السابق .

٢ - يتم الضغط على مفتاح (<---) فيلاحظ تحرك المؤشر الخاص بعمود الاختيارات (Menu Bar) خطوة جهة اليمين . ويلاحظ فتح القائمة الخاصة

بالانشاء (Create) .

٣ - يمكن تحريك المؤشر (Cursor) بواسطة مفتاحي السهم يمين (<---) والسهم شمال (--->) لعرض كل القوائم واختيار القائمة المطلوبة.

٤ - عند وصول المؤشر الى قائمة التجهيز (Set Up) يلاحظ أن الضغط على مفتاح السهم شمال (--->) ضغطة واحدة يؤدى الى الوصول الى آخر قائمة يمين الشاشة وهى قائمة الأدوات (Tools).

٥ - يمكن فتح أي قائمة عن طريق كتابة الحرف الأول من اسم القائمة الموجود على عمود الاختيارات (Menu Bar). فمثلا عند الضغط على الحرف (O) يلاحظ ظهور قائمة (Organize).

٢ - ٢ الاختيار من القائمة

عند فتح أي قائمة يلاحظ أن مؤشر القائمة يكون واقفا على أول اختيار فيها . وبالتالى يظهر هذا الاختيار بالصورة العكسية (Inverse Video)، أي تكون الحروف فاتحة على خلفية قاتمة عكس الوضع الطبيعي الذي يظهر الحروف قاتمة على خلفية فاتحـــة . ولتنفيذ أي اختيار من القائمة المفتوحة يتم تحريك العمود الضوئى (Highlight) لأعلى ولأسفل باستخدام السهمين (↑ ↓) للوصول الى الاختيار المطلوب ثم الضغط على مفتاح الادخال .

ملاحـــظات

١ - لايمكن الاختيار من القائمة بكتابة الحرف الأول من أي اختيار . ولكن يلزم استخدام مفتاحي الأسهم (↑ ↓) والضغط على مفتاح الادخال .

٢ - يلاحظ عند اختيار بعض أوامر اى قائمة ظهور قائمة فرعية أخرى (Submenu) ويتم الاختيار من القائمة الفرعية بنفس الطريقة .

٢ - ٣ عمود الحالة (Status Bar)

وهو عبارة عن عمود ضوئى (Highlight) موجود أسفل الشاشة كما هو موضح بشكل (٢ - ٢) وهو يوضح الحالة التى يتم العمل عليها فى البرنامج حيث يبين الآتى :

- استخدام برنامج المساعد (Assistant) فى التعامل مع البرنامج .
- وحدة الأقراص الجارى العمل عليها .
- ملف قاعدة البيانات المستخدم .
- عدد سجلات الملف ورقم السجل الأول .
- حالة مفاتيح (Num Lock), (Caps Lock), (Ins) علما بأن الوضع المبدئى (Default) لهذه المفاتيح يكون (Lowercase), (Overwrite)

(Alphabetic Mode)، أي الكتابة بدون ازاحة والحروف الصغيرة والحروف الهجائية على الترتيب .

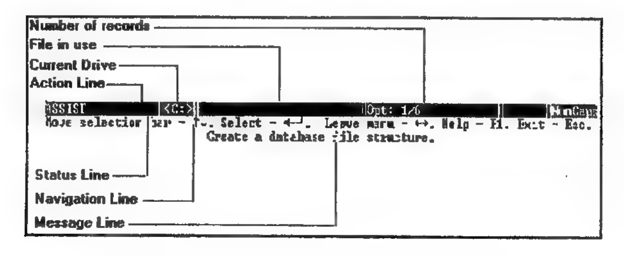
تحذير

عند ظهور كلمة (NUM) على يمين عمود الحالة (Status Bar)، فان مفاتيح الأسهم لا تعمل كمفاتيح أسهم ولكن تعمل في ادخال الأعداد فقط . ويجب الضغط على مفتاح (Num Lock) حتى يمكن تحريك المؤشر باستخدام مفاتيح الأسهم .

ويعرض عمود الحالة معلومات أخرى تعتمد على العمل الجاري تنفيذه . حيث يظهر مثلا معلومات عن السجل الحالي (Current Record) الجاري العمل عليه .

ويلاحظ أيضا أسفل عمود الحالة (Status Bar) وجود سطرين أحدهما يسمى السطر الملاحي (Navigation Line) وهو يوضح المفاتيح المستخدمة لتحريك العمود الضوئي (Highlight) للقائمة المفتوحة . والسطر الثاني يوضح العمل الذي يجري تنفيذه في هذه اللحظة .

كما يلاحظ أيضا وجود سطر آخر فوق عمود الحالة (Status Bar) ، ويسمى خط الأعمال (Action Line) وهو يوضح الأمر المطلوب تنفيذه . كما يوضح شكل الأمر (Syntax) الذي يمكن بواسطته ادخال الأمر مباشرة عن طريق أوامر النقطة (Dot Commands).



شكل (٢ - ٢) عمود الحالة (Status Bar)

٢ - ٤ الغاء الأمر (Cancelling)

يستخدم مفتاح الهروب (Esc) لالغاء أى أمر مهما كان عدد الخطوات والقوائم الفرعية التى تم استخدامها . ويتم الضغط على مفتاح الهروب (Esc) عدة مرات للانتقال من أى قائمة الى القائمة التى تسبقها .

مــلاحظة

عند الضغط على مفتاح الهروب (Esc) من القائمة الرئيسية يتم الخروج من قوائم البرنامج نهائيا . ويتم الانتقال من برنامج المساعد (Assistant) الى مشيرة النقطة (Dot Prompt) وتظهر النقطة (.) فوق عمود الحالة (Status Bar). فاذا أريد العودة الى برنامج المساعد (Assistant) واستخدام القوائم ، يتم كتابة الأمر (ASSIST) والضغط على مفتاح الادخال ، أو يتم الضغط على مفتاح (F2) الذى يؤدى نفس العمل .

٢ - ٥ الحصول على المساعدة (Help)

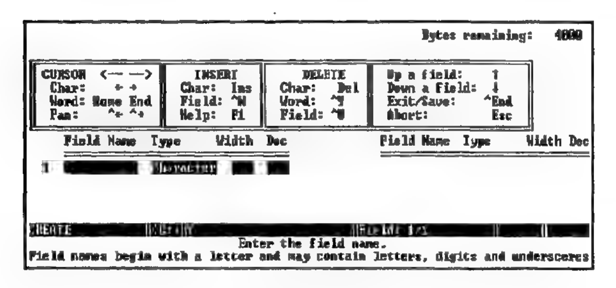
يمكن الحصول على معلومات عن أى أمر من أوامر القوائم عن طريق وضع المؤشر على هذا الأمر والضغط على مفتاح (F1). فى هذه الحالة يظهر مستطيل على الشاشة به معلومات عن هذا الأمر. ثم بالضغط على أى مفتاح يتم الخروج من المساعدة والعودة الى القوائم مرة أخرى .

٢ - ٦ انشاء ملف قاعدة البيانات

لانشاء ملف قاعدة البيانات يلزم أولا تحديد وحدة الأقراص التى يراد تخزين الملف عليها . لذلك يتبع الآتى :

١ - يتم فتح قائمة الأدوات (Tools).
٢ - يتم اختيار الأمر (Set drive) ، ويلاحظ ظهور الأمر (SET DEFAULT TO) أمام مشيرة النقطة (Dot Prompt) الموجودة أسفل الشاشة فوق عمود الحالة (Status Bar) .
٣ - يتم اختيار وحدة الأقراص المراد التخزين عليها والضغط على مفتاح الادخال.
٤ - يتم فتح قائمة الانشاء (Create) واختيار (Database File) ثم الضغط على مفتاح الادخال .
٥ - يتم اختيار وحدة الأقراص المراد تخزين الملف عليها والضغط على مفتاح الادخال .

٦ - يسأل البرنامج عن اسم الملف فيتم كتابته والضغط على مفتاح الادخال .
٧ - يلاحظ ظهور الشاشة المبينة فى الشكل (٢- ٢) والتى عن طريقها يتم ادخال أسماء الحقول (Field Names) وأنواعها وعرضها (Width) وكل حقل يظهر في سطر مستقل . ويتم الانتقال من حقل الى آخر عن طريق تحريك المؤشر باستخدام مفتاحى (↑ ↓) ويمكن ادخال حتى ١٢٨ حقلا باستخدام عدة شاشات .
٨ - يتم كتابة اسم الحقل مع مراعاة أن هذا الاسم لا يزيد عن عشرة حروف ويجب أن يبدأ بحرف ولا يحتوى على أى مسافات بين الحروف . ويمكن استخدام الحروف أو الأرقام أو الشرطة السفلية (Underscore) فى كتابة اسم الحقل .



شكل (٢- ٢) شاشة ادخال مواصفات الحقول (Fields)

٩ - يلاحظ وجود مستطيل أعلى الشاشة يوضح للمستخدم المفاتيح التى يستخدمها لتوجيه المؤشر أثناء الكتابة . وهذا المستطيل يمكن الغاؤه بالضغط على مفتاح (F1) ، كما يمكن اعادته مرة ثانية بالضغط على نفس المفتاح .

ملاحظات

١ - عند انشاء هيكل ملف قاعدة البيانات يمكن الكتابة بالحروف الكبيرة (Uppercase) أو الحروف الصغيرة (Lowercase) أو بالجمع بين الحروف الكبيرة والصغيرة . ويلاحظ فى جميع الأحوال أن الكتابة تظهر على الشاشة بحروف كبيرة (Uppercase) .
٢ - عند اختيار نوع الحقل يمكن كتابة الحرف الأول من النوع . حيث يتم كتابة (C) للحقل الحرفى (Character) ، (N) للحقل العددى (Numeric)

، (D) للحقل التاريخي (Date) ، (L) للحقل المنطقي (Logical) ، (M) لحقل الملاحظات (Memo) . كما يمكن الاختيار أيضا عن طريق الضغط على مسطرة المسافات (Space Bar) حيث يتم التحويل من نوع لآخر مع كل ضغطة عليها.

٣ - عند ادخال حقل تاريخي (Date) يلاحظ أن البرنامج يكتب رقم (8) في خانة عرض الحقل وينتقل المؤشر الى الحقل التالي . وذلك لأن عرض حقل التاريخ ثابت ويساوى ٨ حروف . كما أن الحقل المنطقي عرضه حرف واحد وحقل الملاحظات عرضه (١٠)حروف .

٢ - ٧ تخزين هيكل الملف

يجب تخزين هيكل الملف قبل البدء في ادخال البيانات اليه . ويتم ذلك بالضغط على مفتاح الادخال بعد ظهور رسالة :

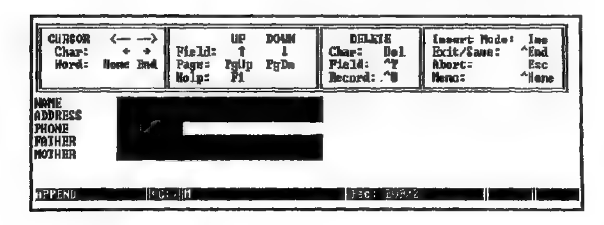
Press Enter to Confirm or Any Other Key to Resume

فتظهر رسالة (Wait) لتوضح أن هيكل الملف جاري تخزينه . وبعد انتهاء عملية التخزين يسأل البرنامج اذا كان المطلوب ادخال بيانات السجلات الآن .

Input Data Records Now ? (Y/N)

فيتم كتابة (Y) لادخال البيانات .

وفي هذه الحالة تظهر شاشة ادخال خالية كما هو موضح في شكل (٢- ٤) بها أسماء الحقول التي تم ادخالها ، وأمام كل اسم عمود ضوئي بنفس عرض الحقل الذي سبق تحديده .



شكل (٢- ٤) شاشة ادخال خالية

٢ - ٨ ادخال البيانات

يتم ادخال البيانات فى الحقول الظاهرة على الشاشة . وينتقل المؤشر من كل حقل الى الحقل الذى يليه . ويجب ملاحظة أن الحروف فى هذه الحالة تظهر كما يتم ادخالها . أى أن الحروف الكبيرة (Uppercase) تظهر كبيرة والحروف الصغيرة (Lowercase) تظهر صغيـرة . وذلك على عكس ما يحدث عند تكوين هيكل الملف .

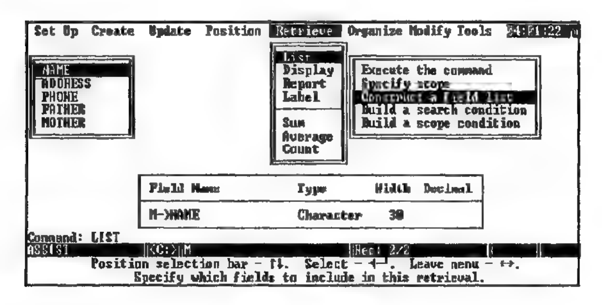
وبعد الانتهاء من ادخال بيانات سجل يظهر السجل التالى . ويتم ادخال بياناته بنفس الطريقة السابقة . وعند الانتهاء من ادخال بيانات جميع السجلات المطلوب ادخالها يتم الضغط على مفتـاحى (Ctrl-End) فى نفس الوقت لتخزين آخـر سجـل تم ادخاله .

ملاحـظة

يتم تخزين كل سجل آليا عند الانتقال الى السجل التالى . أما السجل الأخير فيجب الضغط على مفتاحى (Ctrl-End) لتخزينه .

٢ - ٩ عرض الملف على الشاشة

عندما يراد عرض ملف قاعدة البيانات والبيانات المخزنة فى كل سجل ، يتم اتباع الخطوات التالية : انظر الشكل (٢- ٥)

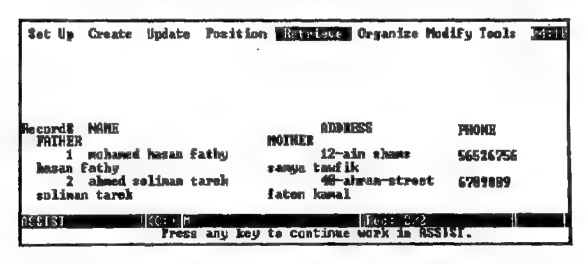


شكل (٢- ٥)

١ - يتم فتح قائمة الاسترجاع (Retrieve) .
٢ - يتم اختيار الأمر (List) من القائمة فيلاحظ ظهور القائمة الفرعية الخاصة بهذا الأمر .
٣ - يتم اختيار الأمر (Construct a Field List) ، يلاحظ ظهور قائمة بجميع حقول الملف الذى سبق انشاؤه فى مستطيل على الجانب الأيسر من الشاشة .
٤ - يتم اختيار الحقول المراد عرض بياناتها وذلك بتحريك المؤشر الموجود فى المستطيل الخاص بالحقول والضغط على مفتاح الادخال عند كل حقل يراد ادخاله ، فيلاحظ ظهور علامة (▶) أمام هذا الحقل . وعندما يراد الغاء حقل سبق اختياره يتم تحريك المؤشر الى هذا الحقل والضغط على مفتاح الادخال مرة ثانية فتختفى علامة (▶) أمام هذا الحقل .
٥ - يتم الضغط على مفتاح السهم يمين (<---) للخروج من هذا المستطيل .
٦ - يتم اختيار الأمر (Execute the Command) .
٧ - يلاحظ ظهور الرسالة التالية :

Direct output to the Printer? (Y/N)

فاذا أريد عرض البيانات على الشاشة فقط يتم الضغط على مفتاح الادخال لأن الوضع المبدئي (Default) هو (N) أى أنه لا يراد طباعتها ولكن عرضها على الشاشة فقط .
٨ - يلاحظ ظهور بيانات الملف على الشاشة كما هو واضح من الشكل (٢- ٦) .
٩ - أما اذا أريد طباعة البيانات على الطابعة فيتم كتابة (Y) ويجب فى هذه الحالة التأكد من أن الطابعة قد تم تشغيلها (ON) وأنها موصلة بالجهاز .



شكل (٢- ٦) عرض بيانات الملف على الشاشة

الباب الثالث

انشاء شاشات الادخال

تم فيما سبق شرح كيفية ادخال البيانات الى ملف قاعدة البيانات باستخدام الشاشة المستخدمة فى البرنامج والتى تماثل هيكل الملف الذى تم تكوينه . وعادة تكون هذه الشاشة غير واضحة للمستخدم كما أن أسماء الحقول تكون غير مفهومة وتحتاج الى كثير من الشرح والتوضيح للمستخدم .

لذلك فان برنامج(+ Dbase III) يتيح تصميم شاشة ادخال واضحة ومفهومة ، وذلك عن طريق مايسمى راسم الشاشة(Screen Painter). ويمكن استخدام هذه الشاشة فى ادخال البيانات وكذلك فى عرض البيانات على شاشة الحاسب .

ويتم تكوين هذه الشاشة عن طريق اتباع الخطوات التالية :

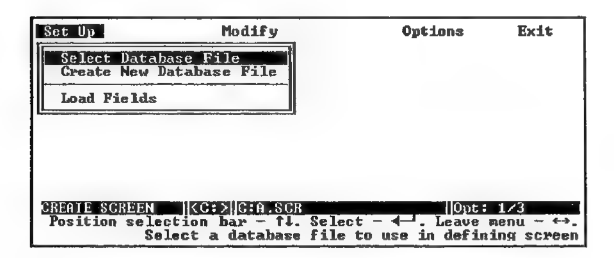
١ - يتم فتح قائمة الانشاء(Create) واختيار(Format) منها.
٢ - يتم اختيار وحدة الأقراص التى يتم فيها تخزين الملف . يلاحظ ظهور الرسالة التالية :

Enter the Name of File

٣ - يتم كتابة اسم ملف شاشة الادخال المطلوبه ، وليس هناك حاجة الى كتابة الامتداد(Extension) لأن برنامج(+ Dbase III) يضيف الامتداد (.scr) آليا.
٤ - يلاحظ ظهور قوائم تصميم شاشة الادخال وسوف يتم شرحها فى الأجزاء التالية .

٣ - ١ قائمة تصميم شاشة الادخال

تحتوى قائمة تصميم الشاشة على القوائم الفرعية المبينة بالشكل(٣- ١)

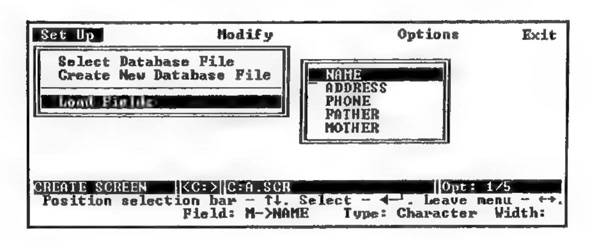


شكل (٣ - ١) قائمة تصميم الشاشة

٢ - ١ - ١ التجهيز (Set Up)

أنظر الشكل (٢- ٢)

وهى قائمة تتيح للمستخدم تحديد اسم ملف قاعدة البيانات المطلوب تصميم الشاشة له. كما تتيح له أيضا اختيار الحقول المطلوب ادخالها فى شاشة الادخال . ويجب التفرقة هنا بين الحقول الفعلية الموجودة فى ملف قاعدة البيانات وبين الحقول المطلوب عرضها فى شاشة الادخال . حيث أنها لا تكون بالضرورة نفس الحقول فى الحالتين . ففى بعض الأحيان قد لا يكون مطلوبا ظهور بعض الحقول أمام القائم بادخال البيانات . و ذلك عندما تكون هذه البيانات سرية مثلا أو يتم ادخالها بصورة مجمعة.

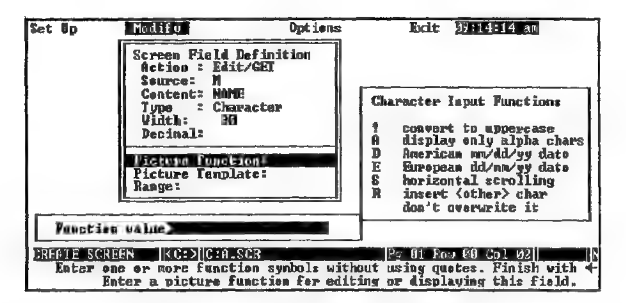


شكل (٢- ٢) قائمة التجهيز (Set Up)

٢ - ١ - ٢ التعديل (Modify)

انظر الشكل (٢- ٣)

وهى قائمة تسمح للمستخدم باضافة حقول جديدة أو تعديل الحقول الموجودة . كما أنها تتيح له تعديل شكل المدخلات فى شاشة الادخال وفى ملف قاعدة البيانات المفتوح .

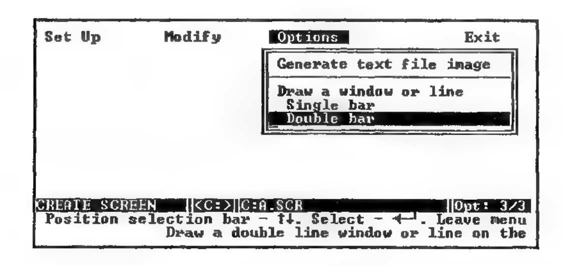


شكل (٢- ٣) قائمة التعديل (Modify)

٣ - ١ - ٣ الاختيارات (Options)

انظر الشكل (٣- ٤)

وهى قائمة تتيح للمستخدم الحصول على نسخة من ملف تعديل الشاشة مكتوبة بشفرة الآسكى(ASCII Code) حتي يمكن تعديلها بواسطة أى برنامج من برامج معالجة الكلمات . كما أنها تتيح له استخدام الخطوط والمستطيلات فى تصميم شاشة الادخال .



شكل (٣- ٤) قائمة الاختيارات (Options)

٣ - ١ - ٤ الخروج (Exit)

وهي قائمة تتيح للمستخدم الخروج من قوائم تعديل الشاشة بعد تخزين التعديلات أو دون تخزينها.

٣ - ٢ خطوات تصميم شاشة الادخال

لتصميم شاشة الادخال يتم اتباع الخطوات التالية :

١ - يتم فتح قائمة الانشاء(Create) واختيار(Format) كما سبق الايضاح ، ثم كتابة اسم الملف المطلوب انشاؤه . فتظهر قوائم تصميم الشاشة كما سبق الايضاح .

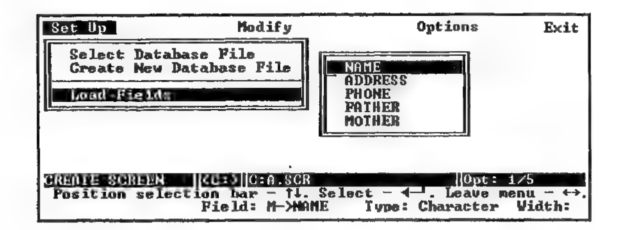
٢ - يكون المؤشر على قائمة التجهيز(Set Up) وبالتالى تكون قائمة التجهيز مفتوحة والعمود الضوئى(Highlight) موجودا على أول اختيار فيها وهو (Select Database File) فيتم الضغط على مفتاح الادخال . وفى هذه الحالة يلاحظ ظهور قائمة فرعية تحتوى على جميع ملفات قواعد البيانات الموجودة على وحدة الأقراص المستخدمة . وهى كل الملفات التى تحتوى على الامتداد (.DBF) . ويظهر عمود ضوئى(Highlight) يتم عن طريقه اختيار الملف المطلوب فتحه ثم الضغط على مفتاح الادخال .

٣ - يتم تحريك المؤشر باستخدام مفتاح السهم لأسفل(↓) الى الاختيار (Load Fields) أى(تحميل الحقول) ثم الضغط على مفتاح الادخال ، فيلاحظ ظهور قائمة فرعية تحتوى على أسماء الحقول الخاصة بملف قاعدة البيانات المفتوح ، فيتم اختيار الحقول المطلوب عرضها فى شاشة الادخال ، ولا يشترط عرض كل حقول ملف قاعدة البيانات كما سبق الايضاح .

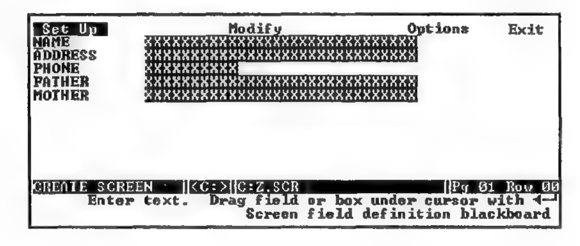
٤ - يتم اختيار الحقول عن طريق تحريك المؤشر على كل حقل مطلوب والضغط على مفتاح الادخال فيلاحظ ظهور علامة(▶) أمام هذا الحقل فيتم الانتقال الى الحقل التالى والضغط على مفتاح الادخال فتظهر العلامة(▶) امامه . ويتم تكرار هذه العملية مع كل الحقول المطلوبة . و يمكن الغاء العلامة امام اى حقل بالضغط على مفتاح الادخال مرة ثانية انظر الشكل (٣- ٥).

٥ - يتم الضغط على مفتاح السهم يمين(<---) أو مفتاح السهم شمال (--->) للخروج من قائمة الحقول . وهذا يعنى أن عملية اختيار الحقول قد انتهت ، فتختفى قائمة الحقول .

٦ - يتم الضغط على مفتاح (F10) فيلاحظ ظهور الحقول التى تم اختيارها أعلى الشاشة . ويلاحظ ظهور اسم كل حقل وأمامه العمود الضوئى(Highlight) الذى يمثل طول هذا الحقل . هذه الشاشة تسمى السبورة(Blackboard) لأنها تسمح للمستخدم برسم شاشة الادخال بالطريقة والشكل الذى يريده كما سيتم الايضاح انظر الشكل(٣- ٦).



شكل (٣- ٥) قائمة التجهيز



شكل (٣- ٦) شاشة السبورة (Blackboard)

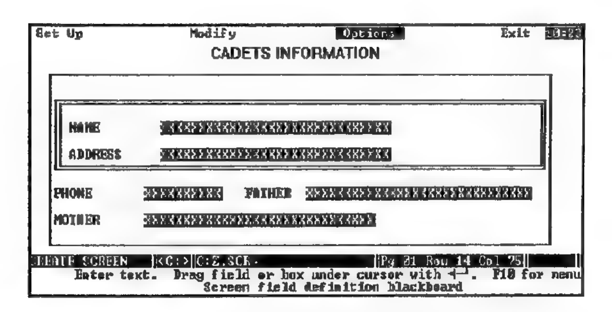
ملاحـظة

يلاحـــظ أن قائـمة التـجهيز(Set Up) تحتـــوى علـــى الاخـــتيار (Create New Database File) وهذا الاختيار يستخدم في حالة انشاء ملف قاعدة بيانات جديد عن طريق شاشة الادخال . وفى هذه الحالة لايلزم انشاء هيكل ملف قاعدة البيانات بالطريقة السابق شرحها.

٢ - ٢ استخدام السبورة (Blackboard)

كما سبق الايضاح ، فان الهدف من تصميم شاشة الادخال هو الحصول على شاشة ادخال للبيانات واضحة ومفهومة للمستخدم . كما أنها يجب أن تكون مقسمة وموزعة بطريقة تسهل للمستخدم ادخال البيانات بسهولة وبسرعة ، لأن الوقت المستهلك فى ادخال البيانات يكون غالبا أكبر من وقت الاسترجاع والتشغيل لهذه البيانات . ولذلك يكون من المهم تقليل هذا الوقت الى الحد الأدنى .

وحتى يتم توضيح عملية تصميم الشاشة يمكن البدء بعرض الصورة النهائية للشاشة بعد عملية التصميم . انظر الشكل (٢- ٧)



شكل (٢- ٧) الصورة النهائية للشاشة بعد عملية التصميم

ومن هذا الشكل يلاحظ الآتى :

١ - تم ادخال عنوان للشاشة .
٢ - تم تنظيم أوضاع الحقول على الشاشة وترتيبها بطريقة منطقية مناسبة للقائم بادخال البيانات .
٣ - تم تغيير أسماء الحقول . هذا مع الاحتفاظ بالأسماء الفعلية للحقول كما هى .
٤ - يلاحظ وجود مستطيلات حول البيانات بخطوط منفردة (Single) أو مزدوجة (Double) .

ولكن كيف يتم تحويل شاشة الادخال الى هذه الصورة ؟

هذا ما سوف يتم ايضاحه فى الجزء التالى .

٣ - ٤ مفاتيح التحكم فى الشاشة

تستخدم بعض مفاتيح لوحة المفاتيح(Keyboard) فى التحكم فى الحقول والكتابة الموجودة فى الشاشة . وهناك وظائف يمكن تنفيذها بعدة وسائل وبأكثر من مفتاح . وفيما يلى توضيح لهذه المفاتيح والوظيفة التى تؤديها مع ملاحظة أن وجود مفتاحين داخل القوس يعنى الضغط على المفتاحين فى نفس الوقت (Simultaneously).

المفتاح	الوظيفة
(Ins) أو(Ctrl-V)	وتستخدم للتحويل بين وضع اضافة الحروف (Insert mode)، ووضع الكتابة مع الغاء الحروف السابقة . و استخدام هذه المفاتيح داخل الحقول يؤدى الى توسيع عرض الحقل (Field Width) وذلك فى الشاشة فقط . بينما يظل عرضه ثابتا فى ملف قاعدة البيانات .
(Ctrl-N)	وتستخدم لاضافة سطر خال جديد بعد مكان المؤشر .
(PgDn) أو(Ctrl-C)	وتستخدم لتحريك الشاشة لأسفل صفحة كاملة(١٨ سطرا)
(PgUp) أو(Ctrl-R)	وتستخدم لتحريك الشاشة لأعلى صفحة كاملة(١٨ سطرا)
مفاتيح الأسهم <--،--> ↑ ↓	وتستخدم لتحريك المؤشر خطوة واحدة فى الاتجاهات الموضحة بالأسهم .
(Home) أو(Ctrl-A)	وتستخدم لتحريك المؤشر الى بداية الكلمة .
(Ctrl -->) أو(Ctrl-B)	وتستخدم لتحريك المؤشر الى نهاية السطر.
(End) أو(Ctrl-F)	وتستخدم لتحريك المؤشر الى بداية الكلمة التالية.
(Enter) أو(Ctrl-M)	وتستخدم لتحريك المؤشر الى بداية السطر التالى. وفى حالة استخدام وضع الاضافة (Insert) فان هذه المفاتيح تستخدم لاضافة سطر خال. واذا تم وضع المؤشر على العمود الضوئى (Highlight) الممثل لحقل من الحقول ، فان هذه المفاتيح تستخدم لسحب هذا العمود الضوئى وتحريكه الى أى مكان . وعند وضع المؤشر على أى مستطيل(Box)،

فان هذه المفاتيح تستخدم في تغيير محيط هذا المستطيل بتكبيره أو تصغيره.

المفتاح	الوظيفة
(Ctrl <--) أو (Ctrl-Z)	وتستخدم لتحريك المؤشر الى بداية السطر.
(Del) أو (Ctrl-G)	وتستخدم لمسح الحرف عند مكان المؤشر . واذا تم استخدامها في حالة وجود المؤشر في العمود الضوئي (Highlight) الخاص بالحقل ، فانها تؤدى الى تقليل عرض الحقل (Width) وذلك في الشاشة فقط ، في حين يظل عرض الحقل في ملف قاعدة البيانات كما هو بدون تغيير.
(Backspace)	و يستخدم لمسح الحرف مكان المؤشر .
(Ctrl-T)	وتستخدم لمسح كل الحروف بدءا من مكان المؤشر حتى بداية الكلمة التالية .
(Ctrl-Y)	وتستخدم لمسح السطر مكان المؤشر.
(Ctrl-U)	وتستخدم لمسح العمود الضوئي (Highlight)الخاص بحقل معين مكان المؤشر . وكذلك تستخدم لمسح مستطيل (Box) يكون المؤشر واقعا على أى نقطة على محيطه.

٣ - ٥ اضافة عنوان للشاشة

تبدأ علمية تصميم الشاشة باضافة عدة سطور خالية فوق الشاشة لادخال العنوان بها. ولتنفيذ ذلك يتم تحريك المؤشر الى أول سطر في الشاشة ثم بالضغط على مفتاحى(Ctrl-N) عدة مرات بعدد السطور المراد ادخالها يلاحظ تحرك جميع الحقول الى أسفل مع اضافة عدة سطور خالية مكانها .

ويمكن تنفيذ هذه العملية أيضا عن طريق الضغط على مفتاح(Ins) للتحويل الى وضع الاضافة(Insert) بدلا من وضع الكتابة مع الالغاء(Overwrite)، ثم الضغط على مفتاح الادخال عدة مرات . يلاحظ في هذه الحالة اضافة عدة سطور خالية مع تحرك الحقول الى أسفل . ثم يتم كتابة عنوان للشاشة في هذه السطور .

تحـــذير

عند الضغط على مفتاح(Ins) لتعديل الحالة . يجب التأكد من عدم وقوف المؤشر على العمود الضوئى الخاص بأى حقل لأنه في هذه الحالة سيؤدى الى زيادة

عرض العمود الضوئي(Highlight) بينما يؤدى الضغط على مفتاح(Del) الى تقليل عرض العمود الضوئى .

ملاحـظة

يـمكن اضافة سطور أخرى بين الحقول بنفس الطريقة باستخدام مفتاحى (Cntrl-N) أو مفتاح الادخال فى وضع الاضافة (Insert).

٣ - ٦ تحريك الحقول (Moving Fields)

لتحريك العمود الضوئى(Highlight) الخاص بأى حقل يتم اتباع الخطوات الآتية :

١ - يتم وضـــع المؤشر على أول العمود الضوئى(Highlight) ويلاحظ أن خط الرسائل (Message Line) الموجود أسفل الشاشة يوضح البيانات الخاصة بهذا الحقل مثل اسم الحقل ونوعه وعرضه .
٢ - يتم الضغط على مفتاح الادخال .
٣ - يتم تحـــريك المؤشر الى أى مكان على الشاشة يراد نقل الحقل اليه باستخدام مفاتيح الاسهم ، ثم يتم الضغط على مفتاح الادخال مرة ثانية ، يلاحظ انتقال العمود الضوئى الممثل لهذا الحقل الى المكان الجديد .
٤ - يتم كتابة عنوان جديد لهذا الحقل أمام هذا العمود الضوئى مع ملاحظة الضغط على مفتاح(Ins) لتحويل الحالة الى وضع الابدال أى الكتابة مع الحذف (Overwrite). ويمكن فى هذه الحالة كتابة أى عنوان واضح للحقل بدلا من الاسم السابق لأن الاسم فى هذه الحالة لايكون مقيدا بشروط معـــينة . فمثـــلا يمكـــن كتابـــة(Telephone Number) بدلا من (T_Num).
٥ - يتم حذف أسماء الحقول السابقة باستخدام مفتاح(Del) مع ملاحظة التحويل الى حالة الكتابة مع الحذف(Overwrite) حتى لا تتحرك الأعمدة الضوئية الخاصة بالحقول .

٣ - ٧ تعديل عرض الحقول (Field Width)

يمكن تعديل عرض الحقول فى الشاشة بطريقتين كالآتى :

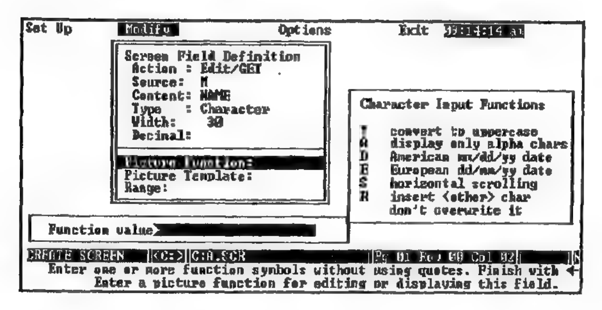
٣ - ٧ - ١ الطريقة الأولى

وهى كما سبق الايضاح تكون عن طريق وضع المؤشر فى أى مكان داخل العمود الضوئى(Highlight) والضغط على مفتاح(Ins) لتوسيع الحقل

أو الضغط على مفتاح(Del) لتقليل عرض الحقل . ويجب ملاحظة أن تغيير عرض الحقل في هذه الحالة يتم بالنسبة للشاشة فقط ولكنه لا يؤثر في عرض الحقل في ملف قاعدة البيانات .

٣ - ٧ - ٢ الطريقة الثانية

وهي عن طريق استخدام قائمة التعديل (Modify) ويتم ذلك كالآتي: أنظر الشكل (٣- ٨) .



شكل (٣- ٨)

١- يتم تحريك المؤشر الى أول العمود الضوئي (Highlight) الخاص بالحقل المطلوب تعديل عرضه .
٢- يتم الضغط على مفتاح(F10) أو مفتاحي(Ctrl-Home) لفتح قائمة التعديل (Modify) .
٣- يلاحظ عند فتح قائمة التعديل ظهور اسم الحقل الذي يقف عنده المؤشر بالاضافة الى نوع الحقل وعرضه .
٤- يتم تحريك العمود الضوئي(Highlight) الخاص بقائمة التعديل (Modify) الي الاختيار(Width) ثم الضغط على مفتاح الادخال ، ويتم تعديل الرقم الموجود في هذه الخانة الى الرقم الجديد ثم الضغط على مفتاح الادخال مرة ثانية .

٥- يتم الضغط على مفتاح (F10) مرة ثانية للخروج من قائمة التعديل . يلاحظ ظهور السبورة(Blackboard) مرة ثانية مع ظهور العمود الضوئي الخاص بالحقل بالعرض الجديد الذى تم ادخاله.

٦- استخدام هذه الطريقة في تعديل عرض العمود يؤدى الى تعديل عرض العمود في كل من الشاشة وملف قاعدة البيانات .

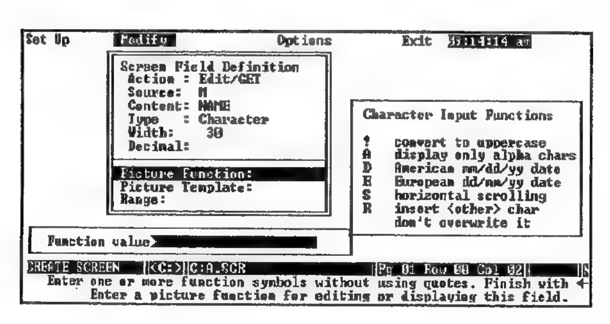
تحـذيـر

عند انقاص عرض حقول ملف قاعدة بيانات سبق ادخال بيانات به ، فان ذلك يؤدى الى اختفاء أى بيانات تزيد عن العرض الجديد . وذلك في حالة الحقول الحرفية ، أما الحقول العددية فان البرنامج يحتفظ بقيمتها ولكنه يظهر حروف(*) مكان أرقام العدد ليوضح أن العدد يزيد عن عرض الحقل المتاح .

٣ - ٨ اضافة حقول جديدة الى شاشة الادخال

يمكن اضافة حقول جديدة الى شاشة الادخال مع اضافتها الى قاعدة البيانات فى نفس الوقت وذلك كالآتى :

انظر الشكل(٣- ٩)



شكل (٣- ٩)

١ - يتم تحريك المؤشر الى أى مكان داخل شاشة الادخال يراد وضع الحقل فيه ، ثم

يتم الضغط على مفتاح(F10) فيلاحظ فتح قائمة التعديل(Modify) ويلاحظ أن القائمة تظهر خالية أي لا تحتوي على أي بيانات مكان اسم الحقل والنوع والعرض . وذلك لأن مؤشر شاشة الادخال كان موجودا في مكان خال وليس في حقل معين .

٢ - يتم تحريك العمود الضوئي(Highlight) الى(Content) ، ثم يتم كتابة اسم الحقل الجديد المطلوب اضافته وليكن مثلا(B_date) أي تاريخ الميلاد. مع ملاحظة أن الاسم هنا يخضع للشروط العامة لأسماء الحقول السابق ذكرها.

٣ - يتم ملء البيانات الاخرى الخاصة بالنوع (Type) ، وعرض الحقل (Width) .

٤ - يتم الضغط على مفتاح(F10) مرة ثانية ، فيتم العودة الى شاشة الادخال ويلاحظ ظهور العمود الضوئي الخاص بهذا الحقل .

٥ - يتم كتابة عنوان لهذا الحقل أمام العمود الضوئي في شاشة الادخال وليكن مثلا (Birth Date) ، مع ملاحظة أن العنوان هنا لايشترط أن يكون هو نفس اسم الحقل ، ولكن يفضل أن يكون عنوانا واضحا للشخص القائم بعملية ادخال البيانات كما سبق الايضاح .

٣ - ٩ مسح حقول من شاشة الادخال

قد يريد المستخدم في بعض الأحيان مسح بعض الحقول التي يراها غير مطلوبة في شاشة الادخال . وقد يريد أيضا مسح بعض الحقول من الشاشة ومن ملف قاعدة البيانات معا . ويتم ذلك باتباع الخطوات التالية :

١ - يتم وضع المؤشر على العمود الضوئي الخاص بالحقل المطلوب مسحه .
٢ - يتم الضغط على مفتاحي (Ctrl-U) .
٣ - يظهر سؤال على الشاشة عما اذا كان المطلوب مسح الحقل من الشاشة فقط أم من الشاشة وملف قاعدة البيانات في نفس الوقت .
٤ - يتم كتابة(N) عندما يراد المسح من الشاشة فقط ، ويتم كتابة(Y) عندما يراد المسح من الشاشة وملف قاعدة البيانات معا.
٥ - يتم مسح عنوان الحقل من الشاشة باستخدام المفتاح(Del) كما سبق الايضاح .

تحذير

عند مسح حقل من شاشة الادخال والملف في نفس الوقت فان أي بيانات سبق تخزينها في هذا الحقل سوف تفقد.

٣ - ١٠ تعديل خصائص الحقل على الشاشة

يتيح برنامج (+ DBase III) للمستخدم التحكم في خصائص الحقل على

الشاشة دون ان يؤثر هذا التحكم على الحقل الفعلي في ملف قاعدة البيانات . حيث يمكن للمستخدم مثلا تحديد مدى معين (Range) للمدخلات . كما يمكنه ايضا ان يسمح للقائم بادخال البيانات بادخال البيان او لايسمح له بذلك حسب الحاجة . ويتم ذلك باستخدام بعض الاختيارات الموجودة فى قائمة التعديل ، و هى الاختيارات الآتية :

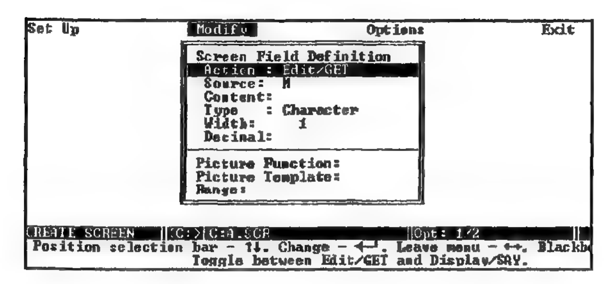
Action , Picture Function , Picture Template , Range

ويتم شرح هذه الاختيارات فى الاجزاء التالية :

٣ - ١٠ - ١ الاختيار (Action) أو الفعل

وهو يعنى الفعل المسموح به للقائم بادخال البيانات في هذا الحقل سواء كان مجرد رؤية بيانات الحقل دون القدرة على تغييرها ، أو تغيير هذه البيانات . ويلاحظ أن الوضع المبدئى لهذا الاختيار يكون (Edit\Get) ومعناه أنه يمكن رؤية بيانات هذا الحقل وتعديلها. ولكى يتم تعديل هذا الاختيار يتم اتباع الخطوات التالية . أنظر شكل (٣-١٠) .

١- من السبورة (Blackboard) يتم تحريك المؤشر حتي يصل الى العمود الضوئى (Highlight) الممثل للحقل المطلوب تعديله .
٢- يتم الضغط على مفتاح (F10) فتظهر قائمة التعديل (Modify) .
٣- يلاحظ ظهور بيانات هذا الحقل فى قائمة التعديل مثل اسم الحقل ونوعه وعرضه .
٤- يتم تحريك العمود الضوئى (Highlight) الى الاختيار (Action) ويلاحظ أن الوضع المبدئي يكون مكتوبا ، وهو (Edit/Get) ومعناه أنه يمكن رؤية بيانات هذا الحقل وتعديلها .
٥ - يتم الضغط على مفتاح الادخال ، فيلاحظ تحول الاختيار الى (Display/Say) وهو يعنى أن بيانات الحقل للعرض فقط ولا يمكن تعديل البيانات أو مسحها.
٦ - يتم الضغط على مفتاح (F10) مرة ثانية للعودة الى السبورة (Blackboard) ويلاحظ اختفاء العمود الضوئى (Highlight) الخاص بهذا الحقل . وهذا يعنى أن هذا الحقل سوف يعرض البيانات الموجودة فقط ولكنه لن يسمح بتعديلها عن طريق شاشة الادخال .



شكل (٣ - ١٠)

٣-١٠-٢ الاختيار (Picture Function) أو دالة الصورة

وهذا الاختيار يسمح للمستخدم بعمل تحويل للمدخلات قبل دخولها الى ملف قاعدة البيانات . فمثلا يمكن تحويل المدخلات الحرفية الى حروف كبيرة (Uppercase) ، بحيث تتحول دائما الى حروف كبيرة بصرف النظر عن الشكل الذى أدخلت به . فيمكن في هذه الحالة أن يقوم القائم بادخال البيانات بادخال المدخلات الحرفية بحروف كبيرة (Uppercase) أو صغيرة (Lowercase) مع دخولها في جميع الأحوال بحروف كبيرة .

ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١ - يتم تحريك المؤشر على السبورة (Blackboard) حتى يصل الى العمود الضوئي (Highlight) الخاص بالحقل المطلوب تعديله .
٢ - يتم الضغط على مفتاح (F10) لاظهار قائمة التعديل (Modify) . ويلاحظ ظهور بيانات الحقل مثل الاسم والنوع والعرض .
٣ - يتم تحريك العمود الضوئي بقائمة التعديل حتى يصل الى الاختيار (Picture Function) ثم الضغط على مفتاح الادخال .
٤ - يلاحظ ظهور رسالة أسفل الشاشة كالآتى :

Function Value

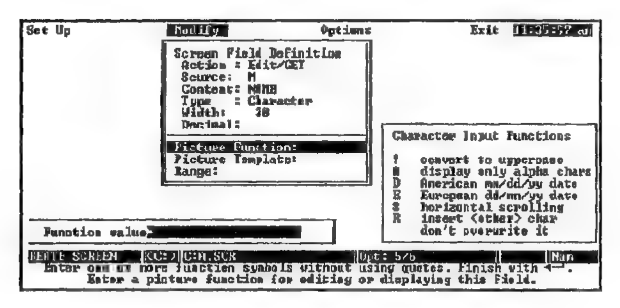
وأمامها عمود ضوئى لادخال القيمة المطلوبة . كما تظهر قائمة على يمين

الشاشة توضح الاختيارات المختلفة لهذا الحقل والتى سيتم شرحها فى الجزء التالى .

٥ - يتم كتابة الحرف (!) فى العمود الضوئى لتحويل الحروف الى حروف كبيرة (Uppercase) والضغط على مفتاح الادخال .

٦ - يتم الضغط على مفتاح (F10) للعودة الى السبورة .

انظر الشكل (٣ - ١١)



شكل (٣ - ١١)

ويلاحظ من قائمة اختيارات دالة الصورة (Picture Function) أن هناك خمسة اختيارات وهى (!) ، (A) ، (R) ، (D) ، (E) ، ويتم شرحها فى الجدول التالى :

!	وهو يؤدى الى تحويل كل الحروف التي يتم ادخالها الى حروف كبيرة .
A	وهو يحول الحروف الى حروف كبيرة (Uppercase) مع ادخال الحروف الهجائية فقط (Alphabet) وعدم قبول أى حروف خاصة (Special Characters) أو اعداد .
D	وهو يؤدى الى ادخال التاريخ على النظام الأمريكى شهر /يوم /سنة (mm/dd/yy) .

E	وهـو يؤدى الى ادخـال التـاريـخ عـلى النظام الأوربـى يوم / شهر/ سنة (dd/mm/yy) .
R	وهو يعمل على الحقول الحرفية مع تحديد شكل معين لها (Template) . حيث يمكن وضع علامات أو حروف خاصة بحيث تعمل كفواصل بين الحروف ولايمكن الكتابة فوقها . فمثلا اذا كانت هناك مدخلات يراد ادخالها مع وجود فواصل خالية بين الحروف . يتم كتابة الشكل الآتى مثلا :

@R A A A A A A A

٣-١٠-٣ الاختيار (Picture Template) أو هيكل الصورة

وهذا الاختيار يسمح للمستخدم بتحديد نوع المدخلات المسموح بها فى هذا الحقل . بمعنى أن الحقل لايقبل أى مدخلات تخالف النوع الذى يتم تحديده من خلال هذا الاختيار. كما يسمح للمستخدم أيضا بتحديد شكل معين (Format) لهذه المدخلات . فمثلا يمكن تحديد شكل رقم التليفون كالآتى :

(XXX) XXX-XXXX

وذلك بالنسبة للدول التى تستخدم هذا الشكل من الأرقام . وهناك عدة اختيارات تظهر أيضا عند استخدام هذا الاختيار يتم توضيحها فى الجدول التالى :

A	وهـي تعنى ادخال الحروف الكبيرة (Uppercase) . وفى هـذه الحـالـة لايقبـل الحقـل أى حـروف صـغيـرة (Lowercase) . وهذا يختلف عن الاختيار (!) الذى يحول الحروف الصغيرة التى يتم ادخالها الى حروف كبيرة .
L	وهـو يسمح فقـط بادخـال المدخلات المنطقية التى تشمل (T, F, Y, N) ولا يقبل أى حرف آخر .
X	وهو يسمح بادخال الحروف التى يمكن أن تشتمل على أرقام أو حروف خاصة (Special Characters) .
#	وهـو يسمـح بادخـال الأعداد التى يمكن أن تشتمل على مسافات وعلامات الجمع (+) وعلامات الطرح (-) .
9	وهو يسمح بادخال الأعداد فقط .
!	وهو يحول الحروف الى حروف كبيرة (Uppercase) .
Other	وهو أى شكل يتم تحديده مثل ###.### (###) .

٣ - ١٠ - ٤ الاختيار (Range) أو المدى

وهو الاختيار الذي يسمح للمستخدم بتحديد حد أدنى وحد أقصى للمدخلات . وفى هذه الحالة فان القائم بادخال البيانات لايمكنه ادخال أى قيم خارج هذا المدى ، لأنها لن تقبل فى هذا الحقل . والقائمة الخاصة بهذا الاختيار تحتوى على الآتى :

Lower Limit	وتعنى أقل قيمة عددية مسموح بها.
Upper Limit	وتعنى أكبر قيمة عددية مسموح بها.

٣ - ١٠ - ٥ اضافة الرسومات الى شاشة الادخال

يتيح البرنامج للمستخدم رسم مستطيلات حول بعض أو كل الحقول تفيد فى توضيح شاشة الادخال أو توضيح أهمية بعض الحقول ، بالاضافة الى جعل هيئة الشاشة مقبولة ومريحة للقائم بادخال البيانات . ويتم ذلك عن طريق اتباع الخطوات التالية :

١- يتم الضغط على مفتاح (F10) للرجوع الى قائمة التعديل (Modify) .
٢- يتم تحريك المؤشر العلوى الى قائمة (Options) التي تسمح برسم نافذة (Window) بخطوط مفردة (Single) أو خطوط مزدوجة (Double) . ثم يتم الرجوع الى السبورة (Blackboard) .
٣- يتم وضع المؤشر على الركن العلوى من اليسار (Upper Left) للمستطيل المطلوب رسمه ، والضغط على مفتاح الادخال .
٤- يتم وضع المؤشر على الركن السفلى من اليمين (Lower Right) للمستطيل المطلوب رسمه ، والضغط على مفتاح الادخال . فيلاحظ رسم مستطيل فى هذا المكان .

ملاحظات

١- المستطيلات أو الخطوط التي يتم رسمها على شاشة الادخال لا تظهر عند طباعة هذه الشاشة على الطابعة كخطوط . بل تظهر غالبا كحروف هجائية (Alphabet) .
٢- اذا أريد توسيع أو تصغير أي مستطيل يتم وضع المؤشر على أى جانب من المستطيل أو علي ركنه والضغط على مفتاح الادخال . ثم يتم نقل المؤشر الى أى نقطة أخري والضغط على مفتاح الادخال مرة ثانية . ويلاحظ تعديل محيط المستطيل .
٣- عندما يراد مسح أى مستطيل يتم وضع المؤشر على أى نقطة على محيطه والضغط على مفتاحى (Ctrl-U) فيلاحظ اختفاء المستطيل .

٣ - ١٠ - ٦ طباعة شاشة الادخال

يمكن طباعة شاشة الادخال عن طريق الضغط على مفتاح (PrtSc) . كما يمكن تخزين الشاشة كملف نص (Text File) وطباعتها في أى وقت باستخدام أوامر نظام التشغيل . ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١ - يتم الضغط على مفتاح (F10) للرجوع الى قائمة رسم الشاشة .
٢ - يتم فتح قائمة الاختيارات (Options Menu) .
٣ - يتم اختيار (Generate Text File Image) .

٣ - ١٠ - ٧ تخزين شاشة الادخال

يتم تخزين شاشة الادخال حتى يتم استخدامها بعد ذلك في ادخال البيانات الى ملف قاعدة البيانات . ويتم تنفيذ ذلك باتباع الخطوات التالية :

١ - يتم الضغط على مفتاح (F10) للرجوع الى قائمة رسم الشاشة .
٢ - يتم فتح قائمة الخروج (Exit) ، واختيار الأمر (Save) .

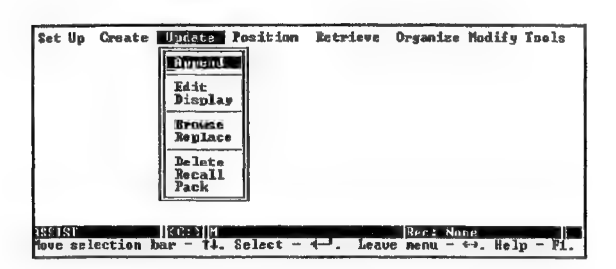
الباب الرابع

تعديل السجــلات
Updating Records

يوضح هذا الباب عمليات عرض البيانات أو تعديلها أو اضافة بيانات جديدة أو حذف بيانات أو اضافة سجلات كاملة أو حذف سجلات كاملة ، وذلك باستخدام قائمة التحديث (Update) ، وهى احدى القوائم الرئيسية الثمانية للبرنامج .

وتحتوى قائمة التحديث (Update) على الاختيارات التالية :

انظر الشكل (٤ -١)



شكل (٤ - ١) اختيارات قائمة التحديث (Update)

١ - الاضافة (Append)

وهو يساعد على اضافة سجلات الى نهاية ملف قاعدة البيانات كما يساعد على تصحيح السجلات التى سبق ادخالها.

٢ - التصحيح (Edit)

وهو يساعد على عرض وتصحيح السجلات الموجودة سجلا تلو الآخر.

٣ - العرض (Display)

وهو يساعد على عرض حتى ١٥ سجلا فى المرة الواحدة.

٤ - العرض مع التصحيح (Browse)

وهو يساعد على عرض وتصحيح واضافة سجلات . وهو يؤدى الى عرض حتى ١٧ سجلا على الشاشة فى المرة الواحدة.

٥ - الاستبدال (Replace)

وهو يساعد على استبدال محتويات حقل معين بمدخلات جديدة فى سجل أو عدة سجلات .

٦ - المسح (Delete)

وهو يساعد على تحديد السجلات المطلوب مسحها ولكنه لا يقوم بمسحها فعليا .

٧ - الاستعادة (Recall)

وهو يساعد على استعادة السجلات التى سبق اعدادها للمسح حتى لايتم مسحها بواسطة الاختيار (Pack) .

٨ - المسح النهائي (Pack)

وهو يساعد على مسح السجلات التى سبق تحديدها بواسطة الاختيار (Delete) .

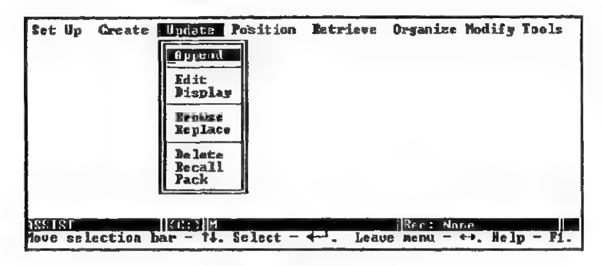
وفى الأجزاء التالية يتم دراسة هذه الاختيارات بالتفصيل .

٤ - ١ الاضافة (Append)

أنظر الشكل (٤ - ٢)

ويستخدم هذا الاختيار عندما يراد اضافة سجلات جديدة بعد آخر سجل سبق تخزينه فى ملف قاعدة البيانات . ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١ - يتم فتح قائمة التحديث (Update)و اختيار (Edit) .
٢ - يلاحظ فتح شاشة الادخال اذا كان قد سبق اختيار شاشة ادخال معينة أو تظهر الشاشة المبدئية (Default) الخاصة ببرنامج(+DBaseIII) .
٣ - يلاحظ وقوف مؤشر صغير على الحقل الأول يتم عن طريقه كتابة بيانات هذا الحقل .
٤ - يلاحظ أن المؤشر ينتقل الى الحقل التالى فى حالة امتلاء الحقل بالبيانات . وفى حالة عدم امتلائه يلزم الضغط على مفتاح الادخال حتى ينتقل المؤشر الى الحقل التالى.



شكل (٤ - ٢) الاضافة

و علاوة علي اضافة سجلات جديدة ، فان هذا الاختيار يتيح عرض السجلات السابقة المخزنة فى الملف وتعديلها أيضا . ولعرض السجلات المخزنة فى الملف يتم استخدام مفتاحي (PgUp) ، (PgDn) . حيث أن مفتاح (PgUp) يعرض السجلات التالية للسجل الحالى ، ومفتاح (PgDn) يعرض السجلات السابقة للسجل الحالى . كما أن مفتاحي (↑ ↓) ينقلان المؤشر الي الحقول المختلفة داخل السجل . أما مفتاحا (<- ، -->) فيستخدمان لتحريك المؤشر داخل الحقل حرفا واحدا فى كل مرة .

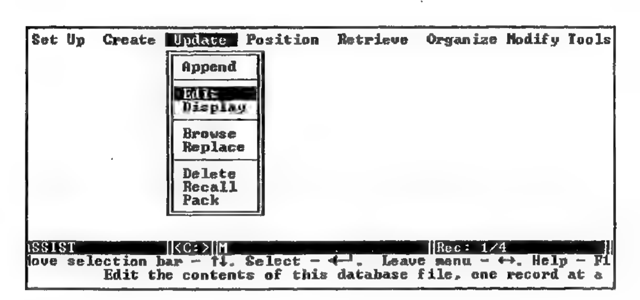
٤ - ٢ التصحيح (Edit)

انظر الشكل (٤ - ٣)

ويستخدم هذا الاختيار عندما يراد تعديل بيانات سجل معين فى ملف قاعدة البيانات . ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١ - يتم فتح قائمة التحديث (Update) واختيار (Edit) .
٢ - يلاحظ فتح شاشة الادخال اذا كان قد سبق اختيار شاشة ادخال معينة أو تظهر الشاشة البدئية (Default) الخاصة ببرنامج (Dbase III+). وذلك بالنسبة للسجل الذي يكون قد سبق تحديده بواسطة الامر (Locate) كما سيتم الايضاح .
٣ - يلاحظ وقوف المؤشر فى الحقل الاول ويتم تعديل البيانات الموجودة به . وفى هذه الحالة يجب التأكد أن البرنامج فى حالة الكتابة مع ازالة الحروف السابقة (Overwrite Mode). وذلك بالتاكد من عدم ظهور كلمة (Ins) على عمود الحالة (Status Bar).

٤ - بعد انتهاء التعديل فى السجل يتم الضغط على مفتاحى (Ctrl-End) لتخزينه .



شكل (٤ - ٢) التصحيح

ملاحظة

في حالة عمل تعديلات في السجل وعدم الرغبة في تخزين هذه التعديلات يتم الضغط على مفتاح الهروب (Esc) بدلا من مفتاحى (Ctrl-End) .

٤-٣ العرض (Display)

ويستخدم هذا الاختيار لعرض حتى ١٥ سجلا في المرة الواحدة . وهذا الاختيار نادرا مايستخدم من قائمة التحديث (Update) لأن هناك امرا آخر فى قائمة الاسترجاع (Retrieve) يؤدى نفس الغرض .

٤-٤ العرض مع التصحيح (Browse)

أنظر الشكل (٤-٤)

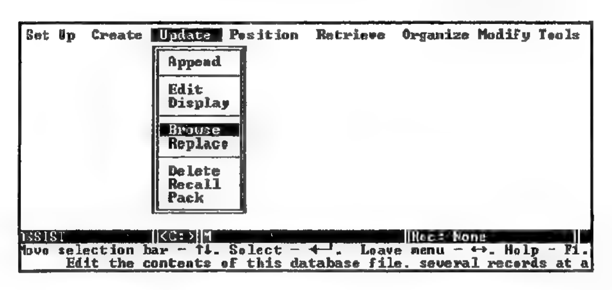
ويستخدم هذا الاختيار عندما يراد عرض السجلات في صورة جدول (Table) مع امكانية التعديل فيها ، وكذلك اضافة سجلات جديدة .

وهذا الاختيار يسمح بعرض حتى ١٧ سجلا فى المرة الواحدة. وكل سجل يظهر

على سطر . وفى حالة احتواء السجل على عدد كبير من الحقول تظهر الحقول الأولى من اليسار بقدر اتساع الشاشة .

ويمكن عرض جميع الحقول باستخدام مفتاحى الأسهم (<-,->) مع مفتاح (Ctrl) لازاحة الشاشة الى اليسار أو الى اليمين على الترتيب .

ولتنفيذ عملية العرض باستخدام الاختيار (Browse) يتم اتباع الخطوات التالية :



شكل (٤-٤) العرض مع التصحيح

١ - يتم فتح قائمة التحديث (Update) واختيار (Browse) .
٢ - يلاحظ ظهور السجلات مع وقوف العمود الضوئى (Highlight) على أحد السجلات . ويكون هذا السجل هو السجل الذى سبق تحديده باستخدام الأمر (Locate) كما سيتم الايضاح فيما بعد . فاذا لم يكن قد سبق تحديده يقف العمود الضوئى على أول سجل فى ملف قاعدة البيانات .
٣ - يلاحظ أن عمود الحالة (Status Bar) يوضح الاختيار الحالى (Browse) ووحدة الأقراص المستخدمة ، واسم الملف المستخدم ، ورقم السجل الحالى الذى يقف عليه العمود الضوئي ، وعدد سجلات الملف .
٤ - يلاحظ وجود مستطيل أعلي الشاشة يوضح المفاتيح التى يمكن استخدامها في التحكم في المؤشر . فاذا أريد اخفاء هذا المستطيل من الشاشة يستخدم المفتاح (F1). واذا أريد اظهاره مرة ثانية يتم الضغط على مفتاح (F1)مرة ثانية أيضا.
٥ - يلاحظ ظهور مؤشر صغير داخل اول حقل في السجل الذي يقف عنده العمودالضوئى . وهذا المؤشر يستخدم في اجراء التعديل المطلوب في هذا الحقل . كما يمكن نقل هذا المؤشر من الحقل الى الحقل الذى يليه باستخدام مفتاح (End). كما يمكن نقل المؤشر الى الحقل الذى يسبقه باستخدام مفتاح (Home).

٦ - لاظهار الحقول المختفية يتم ازاحة الشاشة الي اليسار (Scroll) باستخدام مفتاحى (Ctrl,-->). كما يمكن تحريك الشاشة إلى اليمين باستخدام مفتاحى (Ctrl ,<--) .

٧ - عند الضغط على مفتاح (F10) أو مفتاحى (Ctrl-Home) يلاحظ ظهور قائمة جديدة أعلى الشاشة تتيح للمستخدم بعض الاختيارات التي تفيد في التحكم في الحقول والسجلات المخزنة فى ملف قاعدة البيانات . هذه الاختيارات تكون كالآتى :

١ - القاع (Bottom)
٢ - القمة (Top)
٣ - القفل (Lock)
٤ - رقم السجل (Record No.)
٥ - التجمد (Freeze)
٦ - البحث (Seek)

ويتم شرح خصائص كل اختيار من هذه الاختيارات فى الاجزاء التالية :

٤ -٤- ١ القاع (Bottom)

وهو يؤدى الى ظهور آخر سجل في ملف قاعدة البيانات مع وقوف العمود الضوئى (Higlight) عليه . ويلاحظ في هذه الحاله ظهور رقم هذا السجل في عمود الحالة (Status Bar). كما ان هذا الاختيار يتيح للمستخدم اضافة سجلات جديدة بعد آخر سجل فى الملف . و لتنفيذ ذلك يتم الضغط على مفتاح السهم لاسفل (↓) فيلاحظ ظهور السؤال التالى :

Add new records ? (Y/N)

فيتم كتابة (Y) لاضافة سجل جديد .

٤ - ٤ - ٢ القمة (Top)

وهو يؤدى الي ظهور أول سجل في ملف قاعدة البيانات مع وقوف العمود الضوئى (Highlight) عليه .

٤ - ٤ - ٣ القفل (Lock)

وهو يؤدى الى تثبيت الحقل الموجود فى أقصى يسار الشاشة مع امكانية تحريك باقى الحقول بالنسبة لهذا الحقل . وعند استخدام هذا الاختيار يظهر سؤال على الشاشة عن رقم العمود المراد تثبيت جميع الحقول الموجودة ابتداء منه ويساره .

٤ - ٤ - ٤ رقم السجل (Record No.)

وهو يؤدى الى الوصول الى سجل معين عن طريق رقم هذا السجل . وعند استخدام هذا الاختيار يظهر سؤال عن رقم السجل المطلوب . وفى هذه الحالة يقف العمود الضوئى (Highlight) على هذا السجل .

٤ - ٤ - ٥ التجمد (Freeze)

وهو يؤدى الى حماية كل الحقول من التعديل ما عدا الحقل المطلوب تعديله . وفى هذه الحالة يتم تحديد الحقل المطلوب تعديله حتى يتم حماية باقى الحقول والسماح بتعديل هذا الحقل فقط .

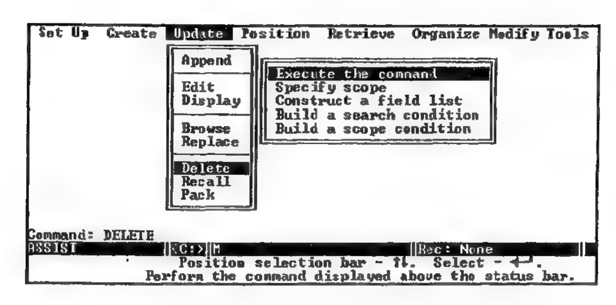
٤ - ٤ - ٦ البحث (Seek)

وهو يؤدى الى البحث عن سجل معين يحتوى على مجموعة حروف (Character String) أو قيمة عددية . وفى هذه الحالة يتم كتابة هذه الحروف أو القيمة العددية حتى يبحث البرنامج عنها فى جميع السجلات . وهذا الاختيار لايظهر الا فى حالة انشاء فهرس (Index) للملف .

٤ - ٥ المسح (Delete)

أنظر الشكل (٤ - ٥)

يستخدم هذا الاختيار لمسح سجل أو سجلات معينة من ملف قاعدة البيانات . وهو فى الواقع لايمسح السجلات ولكنه يقوم بتحديد هذه السجلات حتى يتم مسحها نهائيا باستخدام الاختيار (Pack) . ولتنفيذ هذه العملية يتم اتباع الخطوات التالية :



شكل (٤ - ٥) قائمة المسح

١ - يتم فتح قائمة التحديث (Update) .
٢ - يتم تحريك العمود الضوئي (Highlight) حتى يصل الى الاختيار (Delete) ثم الضغط على مفتاح الادخال .
٣ - يلاحظ فتح قائمة جديدة خاصة بتحديد شروط البحث (Search Conditions) ومدى البحث (Search Scope) .
٤ - يتم تحريك العمود الضوئي الخاص بقائمة البحث حتى يصل الى (Build a Search Condition) والضغط على مفتاح الادخال . يلاحظ ظهور قائمة بأسماء الحقول الموجودة في الملف والتي منها يتم اختيار الحقل أو الحقول التي يتم ادخالها في شروط البحث (Search Conditions) .
٥ - يلاحظ ظهور قائمة بمعاملات المقارنة التى يتم استخدامها فى شروط البحث فيتم اختيار العامل المطلوب .
٦ - يلاحظ خلال ذلك ظهور الأمر الذي يقوم بتنفيذ هذه العمليات فى سطر الأوامر (Command Line) أسفل الشاشة وليكن مثلا :

```
Delete For Age > 25
```

و يعني ذلك مسح سجلات الطلبة الذين تزيد أعمارهم عن ٢٥ سنة .
٧ - يتم الضغط على مفتاح الادخال لادخال هذا الشرط ثم يتم تحريك العمود

الضوئى (Highlight) الخـــاص بقائمـــة البحث حتـــى يصـــل الـــى (Execute the Command) ثم الضغط على مفتاح الادخال .

٨ - يلاحظ ظهور رسالة أسفل الشاشة توضح عدد السجلات التى يتم مسحها. وهى فى الواقع لايتم مسحها نهائيا ولـكن يتم وضع علامات عليها حتى يـتم مـسحها نهائيا بعد ذلك باستخدام الأمر (Pack) .

٩ - للتأكد من وجود علامات امام السجلات المطلوب مسحها ، يتم استخدام الاختيار (Browse) لعرض سجلات الملف . ويلاحظ وجود علامات أمام السجلات التى تم تجهيزها للمسح .

٤ - ٦ الاستعادة (Recall)

ويستخدم هذا الاختيار لاستعادة بعض السجلات التى سبق وضع علامات بها حتى لا يتم مسحها بواسطة الاختيار (Pack) . ويتم تنفيذ ذلك باستخدام نفس الخطوات التى سبق استخدامها مع الاختيار (Delete) لتحديد السجلات التى تحقق شرطا أو شروطا معينة . كما يمكن تحديد سجل معين برقمه والتأكد من اختفاء العلامة أمامه عن طريق استخدام الاختيار (Browse) كما سبق الايضاح .

٤ - ٧ المسح النهائى (Pack)

ويستخدم هذا الاختيار لمسح السجلات التى سبق وضع علامات أمامها تمهيدا لمسحها . حيث أن السجلات التى سبق تحديدها ووضع علامات أمامها تظل موجودة ويمـكن عرضها على الشاشة وتعديلها . وعند استخدام الأمر (Pack)يتم مسحها نهائيا . ولذلك يلزم قبل استخدام الامر(Pack) التأكد من أن السجلات التى تم تمييزها بعلامات لمسحها هى السجلات المطلوب مسحها فعلا .

ولتنفيذ هذه العملية يتم تحريك العمود الضوئى (Hilghlight) حتى يصل الى الاختيار (Pack) والضغط على مفتاح الادخال . يلاحظ فى هذه الحالة ظهور رسالة أسفل الشاشة توضح عدد السجلات التى يتم نسخها . حيث أن الأمر (Pack) يؤدى الى نسخ جميع سجلات ملف، قاعدة البيانات ما عدا السجلات التى تم وضع علامات عليها لمسحها . وفى هذه الحالة يجب اعادة انشاء الفهرس اذا كان قد سبق انشاء فهرس للملف .

الباب الخامس

تنظيم الملف
File Organization

المقصود بتنظيم الملف هو ترتيب السجلات داخل هذا الملف بطريقة تسهل البحث خلاله والوصول الى المعلومات المطلوبة بسرعة وسهولة .

وعند انشاء ملف قاعدة البيانات لأول مرة فان السجلات يتم تخزينها بنفس ترتيب ادخالها أى أن ترتيبها لايعتمد على حقل معين . وعندما يراد البحث عن سجل معين فان البحث دائما يعتمد على محتويات حقل معين مثل البحث عن طالب اسمه فتحى مثلا، أو الطالب الذى يسكن فى عنوان معين ، وهكذا . وفى هذه الحالة يقوم البرنامج بالبحث خلال جميع السجلات ومقارنة بيانات حقل الاسم مثلا بالاسم المطلوب حتى يصل الى السجل الخاص بهذا الطالب . وهذه العملية تأخذ وقتا طويلا نتيجة لأن السجلات غير مرتبة بناء على حقل الاسم بالترتيب الهجائى مثلا . أما اذا كانت مرتبة هجائيا حسب الاسم فان البرنامج يبحث فى ترتيب الحروف حتى يصل الى حرف (F) ثم يبحث عن الاسم المطلوب حسب الترتيب الهجائي للحروف التالية للحرف (F) . وفى هذه الحالة تصبح عملية البحث سهلة وسريعة . وبالمثل يمكن ترتيب السجلات حسب أى حقل آخر عندما يراد البحث عن سجل معين عن طريق بيانات هذا الحقل .

وبرنامج (+ DBase III) يتيح طريقتين لتنظيم الملف أحدهما تسمى الفرز (Sorting) ، والأخرى تسمى الفهرسة (Indexing) .

وفى الأجزاء التالية من هذا الباب يتم القاء الضوء على هاتين الطريقتين وخصائص كل منهما .

٥ - ١ الفرز (Sorting)

الفرز هو طريقة لترتيب السجلات داخل الملف حسب بيانات حقل معين ، وذلك بتغيير المواقع الفعلية للسجلات فى الملف . والطريقة الوحيدة لتنفيذ ذلك هى نسخ الملف بأكمله مع تغيير مواقع السجلات به ، أى أن الفرز يتطلب دائما انشاء ملف جديد . وعند اضافة سجلات جديدة الى الملف الذى تم فرزه ، فان هذه السجلات توضع بعد آخر سجل فى الملف وبالتالى لا توضع فى ترتيبها حسب الحقل الذى تم الترتيب بناء عليه . وفى هذه الحالة يلزم اعادة فرز الملف مرة ثانية . ومع كل فرز جديد يتم انشاء ملف جديد بالاضافة الى الملف الأصلى مما يسبب تحميلا كبيرا (Overload) على ذاكرة الحاسب .

ورغم عيوب الفرز التي سبق ايضاحها الا انه احيانا يكون مطلوبا ، و ذلك عندما يراد مثلا الحصول على ملف مرتب حسب حقل معين بترتيب تنازلى (Descending) بدلا من الترتيب التصاعدى (Ascending) . حيث أن الفرز يتيح الترتيب التنازلى ولكن عن طريق الأوامر (Commands) وليس عن طريق برنامج المساعد (Assistant) كما سيتم الايضاح فيما بعد .

ويجب ملاحظة أن الفرز يغير أرقام السجلات نتيجة نقل السجلات في أماكن أخرى . فمثلا اذا كانت هناك مجموعة من السجلات التي تحتوى على بيانات طلبة ويراد فرزهم أبجديا بالترتيب التصاعدى (Ascending) باستخدام حقل الاسم فان الشكل التالى يوضح الأسماء قبل الفرز وبعده كالآتى :

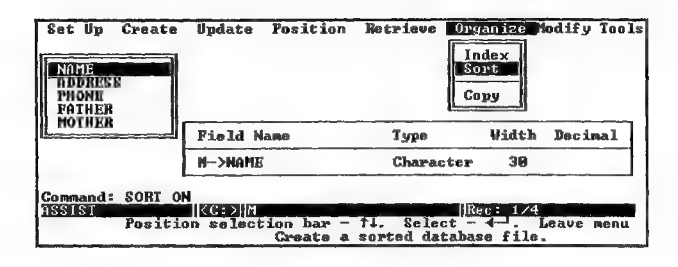
الجدول قبل الفرز			الجدول بعد الفرز		
رقم السجل	الاسم	العمر	رقم السجل	الاسم	العمر
١	محمود	١٨	١	أحمد	١٩
٢	عمر	١٧	٢	بهجت	١٦
٣	أحمد	١٩	٣	سالم	١٠
٤	سالم	٢٠	٤	عمر	١٧
٥	بهجت	١٦	٥	محمود	١٨

يلاحظ من الجدول أن محمود كان رقمه (١) فأصبح (٥) ، وعمر كان رقمه (٢) فأصبح (٤)، وهكذا . أى أن أرقام السجلات تغيرت بعد الفرز .

ولتنفيذ عملية الفرز من خلال برنامج المساعد (Assistant) يتم اتباع الخطوات التالية :

١ - يتم فتح قائمة التنظيم (Organize) والتى تحتوى على ثلاثة اختيارات منها الاختيار (Sort) .
٢ - يتم تحريك العمود الضوئى حتى يصل الى الاختيار (Sort) ، والضغط على مفتاح الادخال.
٣ - يلاحظ ظهور قائمة فرعية تحتوى على أسماء الحقول الموجودة فى الملف والتى يتم منها اختيار الحقول التى يتم الترتيب بناء عليها .
٤ - بعد اختيار الحقول يتم الضغط على مفتاح السهم يمين للخروج من قائمة الحقول . ثم يتم تحديد اسم الملف الذى يتم فرزه . وفى هذه الحالة يمكن كتابة اسم جديد للملف لانشاء ملف جديد مع الاحتفاظ بالملف الأصلى .
٥ - يلاحـظ ظهور رسالة أسفل الشاشة توضح عدد السجلات التى تم فرزها .

أنظرالشكل (٥ - ١)



الشكل (٥ - ١) تنظيم الملف

٥ - ٢ الفهرسة (Indexing)

الفهرسة هي طريقة تستخدم لترتيب السجلات مثل الفرز ، ولكنها تختلف عن الفرز في أنها لاتغير المواقع الفعلية للسجلات . وهي تعتمد على انشاء فهرس مكون من حقلين فقط ، احدهما يحتوى على أرقام السجلات (Record Numbers) والآخر يحتوى على البيانات المطلوب الترتيب بناء عليها مثل الاسم أو الرقم أو .... الخ . وهذا الحقل الآخر يكون مرتبا ترتيبا تصاعديا (Ascending) .

فعندما يراد البحث عن سجل معين بناء على حقل الاسم مثلا يتم البحث فى فهرس الاسم عن هذا الاسم وبالتالى يتم تحديد رقم السجل الخاص به . وعن طريق رقم السجل يمكن الوصول مباشرة الى السجل المطلوب .

ولتوضيح ذلك نفرض أن نفس السجلات المستخدمة فى المثال السابق يراد عمل فهرس لها حسب الاسم فيصبح الفهرس كالآتى :

الملف			الفهرس	
رقم السجل	الاسم	العمر	رقم السجل	الاسم
١	محمود	١٨	٣	أحمد
٢	عمر	١٧	٥	بهجت
٣	أحمد	١٩	٤	سالم
٤	سالم	٢٠	٢	عمر
٥	بهجت	١٦	١	محمود

فعندما يراد الوصول الى السجل الخاص بأحمد مثلا يتم تحديد رقم السجل الخاص به من الفهرس ، وعن طريق هذا الرقم يمكن الوصول الى السجل الخاص به مباشرة . و يلاحظ هنا أن أرقام السجلات تظل كما هي لاتتغير .

وعملية الفهرسة تشبه استخدام فهرس الكتاب للوصول الى موضوع معين . حيث يتم أولا البحث في الفهرس عن هذا الموضوع . وعند الوصول اليه يتم تحديد رقم الصفحة التي تحتوى على هذا الموضوع . وعن طريق رقم الصفحة يمكن الوصول الى الموضوع مباشرة . ورقم الصفحة في هذه الحالة يقابل رقم السجل في ملف قاعدة البيانات .

ولتنفيذ عملية الفهرسة يتم اتباع الخطوات التالية :

١ - يتم فتح قائمة التنظيم (Organize).
٢ - يتم تحريك العمود الضوئي (Highlight) حتى يصل الى الاختيار (Index).
٣ - يلاحظ ظهور الرسالة التالية :

Enter an Index Key Expression

والمقصود هنا الحقل المطلوب استخدامه في الفهرس . ويمكن كتابة اسم هذا الحقل الفهرسي أو اختياره من قائمة الحقول التى تظهر عند الضغط على مفتاح (F10) أو مفتاحي (Ctrl-Home) .
٤ - يمكن استخدام عدة حقول في الفهرس الواحد . وذلك بكتابة اسم الحقل الأول ، ثم علامة الجمع (+) ، ثم اسم الحقل الثانى و..وهكذا . وهذا يعني استخدام الحقل الأول كمفتاح رئيسي والحقل الثانى كمفتاح ثانوى و ..وهكذا. أى أن السجلات يتم ترتيبها حسب الاسم مثلا . وعند تطابق عدة أسماء يتم ترتيبهم حسب العمر مثلا .
٥ - يلاحظ ظهور الرسالة التالية على الشاشة :

Enter the Name of the File

فيتم كتابة الاسم ، ويفضل في هذه الحالة اختيار اسم يوضح نوع الفهرس المستخدم . فمثلا عند انشاء فهرس للأسماء يمكن تسميته (Name) ، مع ملاحظة أن البرنامج يضيف اليه الامتداد (Extension) الذى يكون فى جميع الاحوال (.NDX).
٦ -يمكن انشاء عدة ملفات فهرس (Index Files) للحصول على ترتيب مختلف للسجلات حسب الحاجة . ويمكن استخدام أى نوع من الحقول فى الفهرس ماعدا الحقول المنطقية (Logical) ، وحقول الملاحظات (Memo) . ويمكن جمع عدة حقول فى الحقل الفهرسي (Key Field) ، ولكن يشترط فى هذه الحالة أن تكون جميعها من نفس النوع . فمثلا اذا كان أحد الحقول حرفيا ، فيجب أن تكون باقي الحقول المجموعة عليه حرفية أيضا. وعندما يراد جمع

حقل حرفى مع حقل تاريخي (Date) مثلا ، يجب أولا تحويل حقل التاريخ الى حقل حرفى باستخدام دالة خاصة (Function) تقوم بعملية التحويل .

فمثلا يمكن استخدام الحقل التالى كحقل فهرسى

```
Name + DTOC(Birth_d)
```

حيث تستخدم الدالة (DTOC) لتحويل التاريخ الى حروف . كما سيتم الايضاح فى الجزء الخاص بالدوال (Functions) فى الكتاب الثانى .

٥ - ٢ - ١ استخدام ملف الفهرس

كما سبق الايضاح ، فانه يمكن انشاء أى عدد من ملفات الفهرس المرتبطة بملف قاعدة بيانات واحد . ولكن لا يمكن فتح أكثر من سبع ملفات فهرس في نفس الوقت مع ملف قاعدة البيانات . ويعتبر أول ملف يتم فتحه هو الملف الرئيسي (Master) وباقى الملفات ثانوية . وتتم عملية فتح ملفات الفهرس حسب الخطوات التالية :

١- يتم فتح قائمة التجهيز (Set Up) ، ويكون المؤشر واقفا عند أول اختيار وهو (Database File) فيتم الضغط على مفتاح الادخال .
٢- يظهر سؤال عن وحدة الأقراص المطلوب استخدامها ، فيتم ادخالها.
٣- تظهر ملفات قواعد البيانات الموجودة على وحدة الأقراص المستخدمة فيتم اختيار الملف المطلوب فتحه .
٤- يظهر على الشاشة السؤال التالى :

```
is the file indexed ? (Y/N)
```

٥- يتم كتابة (Y) .
٦- تظهر قائمة بملفات الفهرس التى سبق انشاؤها اذا كان هناك أكثر من فهرس لملف قاعدة البيانات المفتوح .
٧- يتم اختيار الفهرس المطلوب فتحه ، ويلاحظ فى هذه الحالة ظهور كلمة (Master) أمام اسم الملف المفتوح . ويعنى ذلك استخدام هذا الملف كفهرس رئيسى فى ترتيب السجلات . واذا تم استخدام أكثر من فهرس يكون الأول رئيسيا والملفات الباقية ثانوية حسب ترتيبها.

الباب السادس

البحث

Query

عندما يراد استرجاع أى معلومات من قاعدة البيانات ، فان ذلك يتطلب البحث عن السجل الذى يحتوى على هذه المعلومات . ويتم هذا البحث اما بناء على قيمة معينة في الحقول تحقق شروطا معينة ، أو عن طريق رقم السجل الذى يمكن عن طريقه الوصول الى سجل محدد.

٦ - ١ استخدام مؤشر السجلات (Record Pointer)

مؤشر السجلات هو مؤشر منطقى (Logical) يشير الى سجل معين ولكنه لايظهر على الشاشة . وعند اجراء عرض أو تعديل للبيانات تظهر البيانات الخاصة بالسجل الذى يقف عنده المؤشر . وعن طريق توجيه هذا المؤشر يمكن الوصول الى سجل معين .

ولتوجيه المؤشر الى سجل معين تتبع الخطوات التالية :

١ - يتم فتح قائمة المكان (Position) .
٢ - يتم تحريك العمود الضوئى (Highlight) للوصول الى الاختيار (Goto Record).
٣ - يلاحظ أن عمود الحالة (Status Bar) يشير الى السجل رقم (١) وهو الوضع المبدئى (Default) لمؤشر السجلات (Record Pointer).
٤ - يلاحظ ظهور قائمة فرعية تتضمن ثلاثة اختيارات وهى :

Record , Bottom , Top

٥ - عند اختيار(Top) فان مؤشر السجلات (Record Pointer) يظل عند أول سجل فى ملف قاعدة البيانات لأنه يمثل قمة الملف .
٦ - عند اختيار (Bottom) يذهب المؤشر الى آخر سجل فى ملف قاعدة البيانات .
٧ - عند اختيار (Record) يظهر سؤال عن رقم السجل المراد الذهاب اليه فيتم كتابة الرقم والضغط على مفتاح الادخال .
٨ - يتم الخروج من قائمة المكان (Position).
٩ - يمكن الدخول الى قائمة التحديث (Update) واختيار (Edit) ويلاحظ ظهور بيانات هذا السجل الذى تم توجيه المؤشر اليه .
١٠- يمكن تحريك المؤشر عدة خطوات ، وذلك بالرجوع الى قائمة المكان (Position) واختيار (Skip) فيلاحظ ظهور الرسالة التالية :

Enter a Numeric Value :

١١- يتم كتابة عدد السجلات المراد تخطيها أمام هذه الرسالة وليكن (4) مثلا

والضغط على مفتاح الادخال . في هذه الحالة ينتقل المؤشر بعد السجل الذى كان يقف عنده بأربعة سجلات ويقف عند السجل الجديد . فاذا كان المؤشر واقفا فى البداية عند السجل رقم (10) . فانه ينتقل الى السجل رقم (14) .

ملاحظة

للقفز(Skip) عددا من السجلات في الاتجاه العكسى (أى فى اتجاه قمة الملف) يتم استخدام اشارة (-) قبل الرقم المطلوب ادخاله . فمثلا عندما يكون المؤشر أصلا عند السجل رقم (10)وتم كتابة (4-) فان المؤشر ينتقل الى السجل رقم (6) .

٦ - ٢ توجيه المؤشر الى سجل يحقق شروطا معينة

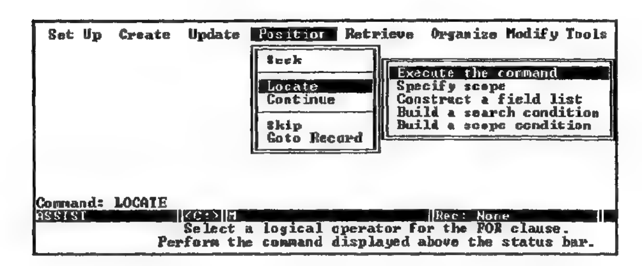
تم في الجزء السابق شرح طريقة توجيه المؤشر الى سجل معين عن طريق رقم السجل (Record Number) . ولكن فى معظم الأحيان يكون رقم السجل غير معلوم . وذلك لأن البرنامج يقوم بتحديد رقم السجل بناء على الترتيب الفعلى لادخال السجلات . وعند حدوث أى تعديل فى هذا الترتيب عن طريق الفرز مثلا (Sorting) ، فان رقم السجل السابق لا يصبح مرتبطا بنفس السجل ، ولكنه يشير الى سجل آخر. أى أن المستخدم لايستطيع متابعة رقم السجل ومعرفة الرقم المقابل لكل سجل . وفى هذه الحالة يلزم الاعتماد على البيانات المخزنة في الحقول ، ووضع شروط معينة للقيم الموجودة في هذه الحقول للوصول الى السجل أو السجلات التي تحقق هذه الشروط . وتوجد عدة طرق لتنفيذ ذلك منها استخدام الأمر (Locate) الموجود فى قائمة المكان (Position) .

٦-٣ استخدام الأمر(Locate)فى الوصول الى سجل محدد

عندما يراد الوصول الى سجل محدد باستخدام الأمر (Locate) فان ذلك يتم عن طريق تحديد قيمة لحقل أو عدة حقول يراد البحث عنها. ويقوم البرنامج بمقارنة هذه القيمة بجميع القيم الخاصة بهذا الحقل أو هذه الحقول لجميع السجلات . وعندما يجد السجلات المطابقة فانه يضع المؤشر عند أول سجل مطابق .

و يمكن استخدام الامر (continue) بعد ذلك للانتقال الى السجل الذى يليه وهكذا . ولتنفيذ ذلك تتبع الخطوات التالية :

١ - يتم فتح قائمة المكان (Position) واختيار (Locate) .
٢ - يلاحظ ظهور قائمة فرعية تحتوى على عدة اختيارات . مع وقوف العمود الضوئى (Highlight) على أول اختيار فى القائمة . انظر شكل (٦ - ١)

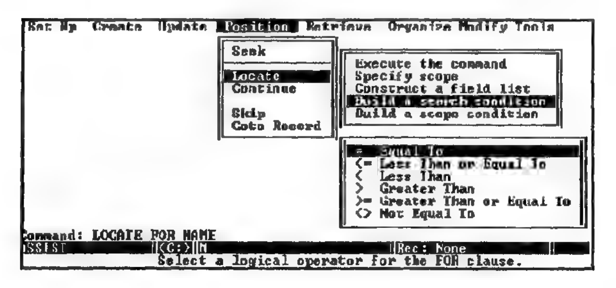


شكل (٦ -١)

٣ - يتم تحريك العمود الضوئي (Highlight) باستخدام مفتاح الاتجاه ↓ حتى يصل الى الاختيار (Build a Search Condition) ، والضغط على مفتاح الادخال .

٤ - يلاحظ ظهور قائمة بأسماء الحقول الموجودة في ملف قاعدة البيانات (DBase File) ، فيتم اختيار الحقل المطلوب استخدامه في تحديد السجل أو السجلات المطلوبة وليكن حقل الاسم (Name) .

٥ - يلاحظ ظهور قائمة بمعاملات المقارنة المطلوب استخدامها في تكوين الشرط المطلوب تحقيقه . انظر الشكل (٦ - ٢)



شكل (٦ - ٢)

٦ - يتم اختيار معامل المقارنة المطلوب وليكن (Equal To =).

٧ - يظهر سؤال عن القيمة المطلوب مقارنتها فيتم ادخالها مع ملاحظة أن القيمة يتم ادخالها بدون علامات تنصيص (Quotation) حتى لو كانت حرفية (String) . و ليكن الاسم المطلوب البحث عنه مثلا هو (Mohamed) . فيتم كتابة الاسم و الضغط على مفتاح الادخال .

٨ - يتم تحريك العمود الضوئي (Highlight) حتى يصل الى الاختيار (No More Conditions) والضغط على مفتاح الادخال .

٩ - يلاحظ ظهور الأمر التالي على خط الأوامر :

```
Locate For Name = Mohamed
```

وهو يمثل الأمر المناظر للاختيارات التي تم تحديدها من القوائم .

١٠ - يتم اختيار (Execute the Command) .

١١ - يلاحظ ظهور رقم أول سجل يحقق هذا الشرط على عمود الحالة (Status Bar). وهذا يعني أن المؤشر يقف الآن عند هذا السجل . فاذا أريد تعديل بيانات هذا السجل تستخدم القوائم فى ذلك كما سبق الايضاح .

١٢ - اذا أريد الوصول الى سجل آخر يحقق الشرط ، يتم تحريك العمود الضوئي (Highlight) الى الاختيار (Continue) والضغط على مفتاح الادخال ، فيلاحظ تغير رقم السجل المكتوب فى عمود الحالة (Status Bar) . وهكذا يمكن الوصول الى جميع السجلات التى تحقق الشرط .

١٣ - يمكن استخدام عدة شروط فى البحث عن السجل بالربط بين الشروط بواسطة المعاملات المنطقية (Logical Operators) مثل (OR , AND) التي تظهر فى القائمة الفرعية .

ملاحظة

يمكن استخدام معاملات المقارنة مثل أكبر من (>) ، اصغر من (<) مع المدخلات الحرفية (String) . وفي هذه الحالة يتم مقارنة الحرف الأول فى القيمتين حسب ترتيبه في الترتيب الهجائي للحروف .

٦ - ٤ استرجاع السجلات (Retrieving)

يمكن عن طريق قائمة الاسترجاع (Retrieve) عرض بيانات عدد من السجلات التى تحقق شرطا أو شروطا معينة . ويستخدم لذلك الامر (List) ، والامر (Display) .

ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١ - يتم فتح قائمة الاسترجاع (Retrieve) واختيار الامر (List) .
٢ - يتم اختيار الامر (Construct a Field List) .
٣ - يلاحظ ظهور قائمة بحقول الملف ، فيتم اختيار الحقول المراد عرض بياناتها .
٤ - يتم الضغط على مفتاح السهم يمين (< - -) للخروج من قائمة الحقول .
٥ - يتـم اختيـار (Build a Search Condition) ، ويتم اختيار الحقل المطلوب استخدامه فى شرط البحث .
٦ - يلاحظ ظهور قائمة معاملات المقارنة .
٧ - يتم اختيار العامل المطلوب والضغط على مفتاح الادخال .
٨ - يلاحظ ظهور سؤال عن القيمة المطلوب مقارنتها . فيتم ادخالها والضغط على مفتاح الادخال .
٩ - اذا أريد ادخال شرط آخر يتم اختيار العامل المنطقى المطلوب استخدامه اذا كان (AND) أو (OR) .
١٠ - يتم اختيار الامر (Execute the Command) .
١١ - يلاحظ ظهور السؤال التالى :

Direct The Output To The Printer ? (Y/N)

فاذا أريد عرض السجلات على الشاشة فقط يتم كتابة (N) ، أما اذا أريد طباعة هذه السجلات فيتم كتابة (Y) .

ملاحظة

يمكن استخدام الاختيار (Display) بدلا من الاختيار (List) لتحقيق نفس النتيجة . والفرق بينهما أن (List) فى الوضع المبدئى له يؤدى الى عرض جميع سجلات ملف قاعدة البيانات . أما الاختيار (Display) فان الوضع المبدئى له يؤدى الى عرض سجل واحد فقط ، وهو السجل الذى يقف عنده المؤشر . ومع ذلك فان ادخال شروط معينة فى الحالتين يؤدى الى الوصول الى نفس النتيجة .

الباب السابع

ملفات البحث
Query Files

عندما يريد المستخدم استرجاع مجموعة محددة من السجلات التي تحقق نفس الشروط (فمثلا عندما يراد دائما البحث خلال سجلات الموظفين الذين التحقوا بالعمل ابتداء من سنة ١٩٨٠ وتزيد أعمارهم عن ٣٠ سنة) . فبدلا من تكرار ادخال هذه الشروط عند كل عملية بحث عن أى موظف ، فمن الأفضل انشاء ملف بحث (Query File) و ادخال كل الشروط المطلوبة فيه . ويستخدم هذا الملف كمرشح (Filter) يتم من خلاله تصفية قاعدة البيانات وعدم السماح بالمرور من هذا المرشح الا للسجلات التى تحقق الشروط الموجودة به . ويمكن تخزين هذا الملف واستخدامه وقت الحاجة . كما يمكن انشاء عدة ملفات بحث واستخدام أى ملف منها مع ملف قاعدة البيانات . وهذه الطريقة تتيح للمستخدم مرونة كاملة فى التعامل مع السجلات .

٧ - ١ انشاء ملف البحث

الهدف من ملف البحث كما سبق الايضاح هو تصفية عدد السجلات التى يتم عرضها والتعامل معها وذلك عن طريق استبعاد السجلات التى لاتحقق شروطا معينة . ويتم ادخال هذه الشروط عن طريق نموذج خاص كما هو موضح بالشكل (٧ - ١) ، وعن طريق اتباع الخطوات التالية :

```
Set Filter          Nest          Display          Exit
 Field Name
 Operator
 Constant/Expression
 Connect
 Line Number        1

Line | Field | Operator | Constant/Expression | Connect
1
2
3
4
5
6
7

Position selection bar - ↑↓.  Select - ←┘.  Leave menu - ←→.
Select a field name for the filter condition.
```

شكل (٧ - ١)

١- يتم فتح قائمة التجهيز (Set Up) واختيار ملف قاعدة البيانات (DBase File) المطلوب فتحه .

٢ - يظهر سؤال عما اذا كان الملف تم فهرسته (Indexed) أم لا . وفى حالة فهرسة الملف يتم كتابة اسم ملف الفهرس (Index File) .

٣ - يتم الخروج من قائمة التجهيز (Set Up) عن طريق مفتاح السهم يمين (-->) .

٤ - يتم فتح قائمة البحث (Query) ، ويتم تحديد وحدة الأقراص الموجود بها القرص المطلوب تخزين هذا الملف به .

يلاحظ ظهور عمود الاختيارات الخاصة بالبحث . وهذا العمود يحتوى على اختيارات يتم عن طريقها تحديد شروط البحث . كما يلاحظ ظهور جدول (Table) يتم فيه وضع الحقول والقيم المطلوب مقارنتها ومعاملات المقارنة . وهذا الجدول يتيح ادخال عدة شروط كما يتيح الربط بين هذه الشروط باستخدام المعاملات المنطقية (AND) ، (OR) ، لتكوين شروط مركبة .

ولتمثيل شرط مركب في هذا الجدول يتم أولا كتابة هذا الشرط خارج الجهاز . فمثلا اذا أريد ادخال شرط يحدد الموظفين الذين التحقوا بالعمل بعد ١-١- ١٩٩٠ وتزيد أعمارهم عن ٣٠ سنة أو تزيد مرتباتهم عن ٣٠٠ جنيه بشرط أن يكونوا متزوجين ، يتم كتابة هذا الشرط خارج الجهاز كالآتي :

```
                                 ┌ AGE > 30     ┐
DAT_ENT > 1/1/1990 AND           │    OR        │     AND MARRIED
                                 └ SALARY>300   ┘
```

وذلك بفرض أن (DAT_ENT) يمثل اسم الحقل الخاص بتاريخ التحاق الموظف ، و (AGE) يمثل اسم الحقل الخاص بعمر الموظف ، و (SALARY) يمثل مرتب الموظف و (MARRIED) هو اسم الحقل المنطقى (Logical) الذى يوضح اذا كان الموظف متزوجا أو غير متزوج .

ولكتابة هذا الشرط المركب بالصورة التى يميزها برنامج (DBase III +) يصبح كالآتي :

DAT_ENT>1/1/1990 AND ((AGE > 30) OR (SALARY >300))AND MARRIED.

ويلاحظ هنا استخدام الأقواس (Parentheses) لتحديد ترتيب تنفيذ البرنامج لعمليات المقارنة .

ولكتابة هذا الشرط بالجدول تتبع الخطوات الآتية : أنظر الشكل (٧ - ٢)

١ - عند ظهور عمود الاختيارات (Menu Bar) الخاص بقائمة البحث (Query) يلاحظ وقوف المؤشر عند أول اختيار فى القائمة وهو (Set Filter) وبالتالي فتح القائمة الخاصة به . ويلاحظ كذلك وقوف العمود الضوئي (Highlight) عند أول اختيار فى القائمة وهو (Field Name) فيتم الضغط على مفتاح الادخال .

```
Set Filter            Nest                 Display             Exit
 Field Name                                  NAME
 Operator                                    ADDRESS
 Constant/Expression                         PHONE
 Connect                                     FATHER
                                             MOTHER
 Line Number          1

 Field Name           Type     Width  Decimal   ant/Expression   Connect
 M->NAME              Character  30
 3
 4
 5
 6
 7
    Position selection bar - ↑↓.  Select - ◄─┘.  Leave menu - ↔.
           Select a field name for the filter condition.
```

شكل (٧ - ٢)

٢ - يلاحظ ظهور قائمة بأسماء الحقول الموجودة بالملف فيتم اختيار الحقل الأول فى الشرط وهو (Name) فيلاحظ كتابة اسم هذا الحقل فى عمود الحقل (Field) فى الجدول .

٣ - يتم تحريك العمود الضوئى الى الاختيار (Operator) فيلاحظ ظهور قائمة بمعاملات المقارنة (Relational Operators) . فيتم اختيار المعامل المطلوب وهو (> More Than) فيلاحظ ظهور هذا العامل فى عمود العامل (Operator) فى الجدول . انظر الشكل (٧ - ٣)

```
Set Filter            Nest                 Display             Exit
 Field Name           NAME
 Operator
 Constant/Expression
 Connect

 Line Number          1

 Line  Field     Operator        Constant/Expression     Connect
 1     NAME
 2
 3
 4
 5
 6
 7
    Position selection bar - ↑↓.  Select - ◄─┘.  Leave menu - ↔.
        Select a comparison operator for the filter condition.
```

شكل (٧ - ٣)

٤ - يتم تحريك المؤشر الضوئى الى الاختيار (Constant/Expression) وكتابة التاريخ كالآتى :

CTOD ("01/01/1990")

ويلاحظ هنا استخدام الدالة (CTOD) لتحويل التاريخ الى قيمة يميزها البرنامج . وهذا سوف يتم ايضاحه فيما بعد.

٥ - يتم تحريك المؤشر الضوئي الى الاختيار (Connect) وكتابة العامل المنطقي المطلوب للربط بين هذا الشرط والشرط التالى له . وفى هذه الحالة يتم كتابة المعامل المنطقى (AND) .

٦ - يتم اضافة الشروط الأخري بنفس الطريقة مع الربط بينها بالمعامل المنطقى المطلوب .

٧ - يتم الضغط على مفتاح (<--) للخروج من قائمة (Set Filter) ، ويلاحظ انتقال مؤشر عمود الاختيارات (Menu Bar) الى الاختيار التالى وهو (Nest) والذى يسمح باستخدام شروط مركبة عن طريق الأقواس كما سيتم الايضاح فى الجزء التالى .

٧ - ٢ تداخل الشروط (Nesting)

عندما يبحث برنامج (DBase III+) عن سجل محدد بناء على شروط معينة متداخلة ، فانه يتبع قواعد الأسبقية (Precedence Rules) المعروفة فى معظم لغات الحاسب لتحديد ترتيب تنفيذ هذه الشروط .

وعن طريق استخدام الأقواس يمكن التحكم فى اولويات تنفيذ الشروط فى العلاقة . لذلك يتم وضع الأقواس حول كل شرط يراد تنفيذه قبل الشروط الأخرى .

ففى المثال السابق تم وضع العلاقة بالصورة التالية :

```
DAT_ENT > 1/1/1990 AND ((AGE> 30) OR (SALARY> 300))
AND MARRIED
```

ولاضافة الأقواس فى العلاقة تتبع الخطوات التالية :

١ - يتم فتح قائمة (Nest) فيلاحظ وقوف العمود الضوئى (Highlight) الخاص بها عند الاختيار (Add) فيتم اختيار (Start) وكتابة رقم السطر الذى يتم وضع أول قوس عنده ، ثم اختيار (End) وكتابة رقم السطر الذى يتم كتابة نهاية القوس عنده . ويلاحظ ظهور الأقواس على الجدول .

٢ - يمكن اتباع نفس الخطوات لكتابة أى أقواس أخرى خارجية .

٧ - ٣ عرض وتخزين ملف البحث (Query File)

بعد ادخال شروط البحث في الجدول كما سبق الايضاح يلزم أولا التأكد أن هذه الشروط سوف تؤدى الى اختيار السجلات المطلوبة . ويتم ذلك عن طريق الاختيار (Display) من قائمة الاختيارات (Menu Bar) . ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١ - يتم استخدام مفتاح السهم يمين (-->) للانتقال الى الاختيار (Display) والضغط على مفتاح الادخال .
٢ - يلاحظ ظهور بيانات أول سجل يحقق شروط البحث .

انظر الشكل (٧ - ٤)

```
Set Filter                 Nest                 Display              Exit

AME        mohamed hasan fathy
DDRESS     12-ain shams
HONE       56526756
ATHER      hasan fathy
OTHER      samia tawfik

Line | Field    | Operator     | Constant/Expression | Connect
1    | NAME     | Begins with  | "m"                 | .OR.
2    | ADDRESS  | Matches      | "ain shams"         |
3
4
5
6
7

Rec: 1/2
Next/Previous record - PgDn/PgUp.  Toggle query form - F1.  Leave option -
        Display records in the database that meet the query condition.
```

شكل (٧ - ٤)

ملاحـــظة

عند وجود أى خطأ فى جدول البحث تظهر الرسالة التالية :

Invalid Filter

ولايتم ظهور أي سجلات . وفى هـــذه الحـــالة يتم الرجـــوع الى قائمة تجهيز المرشح (Set Filter) وتصحيح الأخطاء الموجودة قبل تخزين ملف البحث .

٣ - يمكن الضغط على مفتاح (PgDn) لعرض السجل التالى والذى يليه وهكذا .
٤ - بعد عرض عدة سجلات والتأكد أنها تحقق الشروط يتم الضغط على مفتاح السهم

يمين (<---) للخروج من قائمة العرض (Display) وفتح القائمة التالية لها وهي قائمة الخروج(Exit) .

٥ - من قائمة الخروج يتم اختيار الأمر (Save) ، فيتم تخزين ملف البحث بنفس الاسم الذى سبق تحديده . انظر الشكل (٧ - ٥)

Set Filter Nest Display Exit 10:43:23 am

Save
Abandon

Line	Field	Operator	Constant/Expression	Connect
1	NAME	Begins with	"a"	.OR.
2	ADDRESS	Matches	"ain shams"	
3				
4				
5				
6				
7				

CREATE QUERY <C:> E:N.QRY Opt: 1/2

Position selection bar - ↑↓. Select - ◄┘. Leave menu - ↔.
Exit and save changes.

شكل (٧ - ٥)

٧ - ٤ استخدام ملف البحث

عندما يراد استخدام ملف البحث يتم فتحه من خلال قائمة التجهيز (Set Up) . حيث يتم اختيار اسم قاعدة البيانات أولا ثم اختيار ملف البحث الذي سبق انشاؤه . ويمكن انشاء عدة ملفات بحث وتخزينها ثم اختيار ملف البحث المطلوب فى كل مرة يتم فيها فتح ملف قاعدة البيانات (DBase File) .

٧ - ٥ المعاملات الحرفية (Character Operators)

كما سبق الايضاح فانه عند كتابة معاملات المقارنة ، فان ذلك يتم لكل حقل يتم اختياره من ملف قاعدة البيانات (DBase File) . ولذلك فان ما يظهر من هذه المعاملات هو المعاملات التى تخص نوع الحقل المستخدم سواء كان عدديا أو حرفيا أوتاريخيا أو ... الخ. ويقوم المستخدم باختيار المعامل المطلوب ادخاله في الجدول . وبالنسبة لمعاملات المقارنة العددية فهي معروفة ولا تحتاج الى شرح . أما معاملات المقارنة الحرفية فيتم شرحها فى الجدول التالى :

المعامل	الوظيفة
= Matches	وهو يعني أن الحروف الموجودة في الحقل تماثل تماما الحروف الموجودة في الثابت الحرفي المستخدم في المقارنة متضمنا حالة الحروف اذا كانت كبيرة (Upercase) أو صغيرة (Lowercase)
<> Does not match	وهو يعني أن الحروف في الحقل لاتماثل الحروف الموجودة في الثابت الحرفي . وفي هذه الحالة يتم اختيار السجلات التي لا يطابق حقل معين فيها مقداراً ثابتا معينا (Constant) . فمثلا اذا أريد استرجاع كل السجلات الخاصة بالموظفين من جنسيات غير مصرية مثلا يتم ادخال الشرط كالآتي : Nation <> Egypt
= Begins With	وهو يعني أن أول حرف أو مجموعة من الحروف في حقل معين تماثل الحرف أو الحروف المطلوب مقارنتها.
End With	وهو يعني أن آخر حرف أومجموعة من الحروف في حقل معين تماثل الحرف أو الحروف المطلوب مقارنتها.
$ Contains	وهو يعني أن الحقل يحتوى على حروف معينة (في أي مكان داخله) . فمثلا يمكن البحث عن السجلات الخاصة بالموظفين الذين يسكنون بحي شبرا وذلك عن طريق البحث عن كلمة شبرا في حقل العنوان (Address) .
Does not contain	وهو يعني أن الحقل لايحتوى داخله على حروف معينة. وهو عكس المعامل السابق .
Is contained in	وهو يعني أن الحقل الحرفي موجود ضمن مجموعة معينة من الحروف .
Is not contained in	وهو يعني أن الحقل الحرفي غير موجود ضمن مجموعة معينة من الحروف .
> Comes after	وهو يعني أن أول حرف في الحقل الحرفي يأتي بعدحرف معين في الترتيب الهجائي .
> = Comes after or matches	وهو يعني أن أول حرف في الحقل الحرفي يماثل أو يأتي بعد حرف معين في الترتيب الهجائي .

الباب الثامن

التقارير والعناوين المختصرة

Reports and Labels

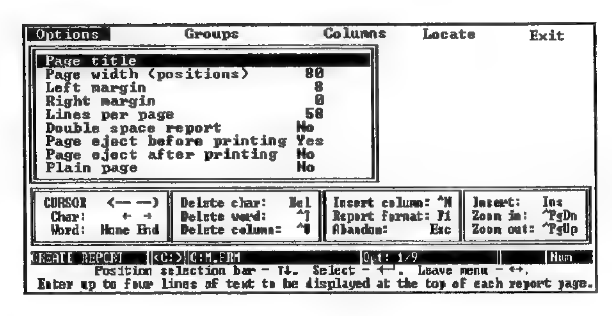
يحتاج المستخدم الى تقارير مكتوبة متضمنة بيانات من بعض السجلات ، كما يحتاج في بعض الأحيان الى عناوين مختصرة(Labels) تحتوى على بيانات سريعة من أى سجل مثل الاسم والعنوان والتليفون . ولتنفيذ ذلك يلزم أولا انشاء ملف التقرير أو العناوين المختصرة واستخدامه بعد ذلك في كتابة بيانات أى سجل أو مجموعة من السجلات .

٨ - ١ انشاء ملف التقرير (Repoart File)

قبل البدء في انشاء ملف التقرير ، يجب اولا التاكد من فتح ملف قاعدة البيانات المستخدم (Database File) حتـــى يتم اختيار الحقول المطلوب ظهورها في التقرير . وكما سبق القول في انشاء ملفات البحث(Query Files) ، يمكن انشاء عدة ملفات تقارير(Report Files) ، ثم اختيار ملف التقرير المطلوب استخدامه وقت الحاجة .

ولانشاء ملف التقرير يتم اتباع الخطوات التالية :

أنظر الشكل (٨ - ١)



شكل (٨ - ١)

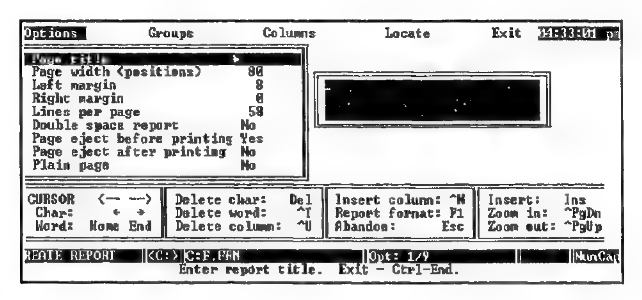
١ - يتم فتح قائمة الانشاء(Create) واختيار (Report) .
٢ - يتم اختيار وحدة الأقراص المطلوب تخزين الملف فيها.
٣ - يتم كتابة اسم ملف التقرير المطلوب انشاؤه . مع ملاحظة أن البرنامج يضيف الامتداد (FRM.) .
٤ - يلاحظ ظهور عمود الاختيارات(Menu Bar) مع وقوف مؤشر هذا العمود على أول اختيار وهو الاختيار(Options) مع فتح القائمة الخاصة به .
٥ - يلاحظ وجود عدة اختيارات خاصة بعنوان صفحة التقرير(Page Title) وأبعاد الصفحة وشكل الكتابة ... وهكذا .
٦ - يلاحظ وجود قائمة مساعدة(Help) أسفل الشاشة لمساعدة المستخدم عند كتابة عنوان التقرير والبيانات المختلفة . وذلك عن طريق توضيح وظائف الأسهم المختلفة التي يتم عن طريقها تحريك مؤشر الكتابة . ولاخفاء هذه القائمة يتم الضغط على مفتاح(F1) ، كما يمكن الضغط عليه مرة ثانية لعرض قائمة المساعدة(Help) عند الحاجة الى ذلك . ويلاحظ أيضا ظهور مساحة خالية مكان هذه القائمة تسمى(Report Format) تظهر بالتبادل مع قائمة المساعدة عند الضغط على مفتاح (F1) .

٨ - ١ - ١ عنوان التقرير

يسمح البرنامج بكتابة عنوان للتقرير حتى أربعة سطور ، ويتم ذلك عن طريق الآتى :

مع وجود العمود الضوئي(Highlight) عند الاختيار (Page Title)يتم الضغط على مفتاح الادخال . ويلاحظ ظهور مستطيل يمكن عن طريقه ادخال العنوان المطلوب . ويظهر مؤشر صغير ليساعد على الكتابة . وعند الانتهاء من كتابة عنوان التقرير يتم الضغط على مفتاح الادخال عدة مرات حتى يصل المؤشر الى السطر الأخير ، ثم يتم الضغط على مفتاح الادخال مرة أخرى حتى يتم ادخال العنوان .

أنظر الشكل (٨ - ٢)



شكل (٨ - ٢)

٨ - ١ - ٢ التحكم فى شكل الصفحة (Page Format)

تستخدم باقى الاختيارات في قائمة(Options) في تحديد شكل الصفحه ، مــع ملاحظة أن القيم الموجودة أمام كل اختيار هي القيم المبدئية (Default) ، وهى تكون في معظم الأحيان مناسبة ولا تحتاج الى تغيير . ولتغيير أى اختيار يتم تحريك العمود الضوئى(Highlight) الى هذا الاختيار والضغط على مفتاح الادخال ثم كتابة الرقم الجديد. أو يتم الضغط على مفتاحى (↑ ↓) لزيــادة الرقــم المكتوب أو انقاصه بمقدار (١) مع كل ضغطة .

وفى الجزء التالي يتم توضيح اختيارات هذه القائمة والقيم المبدئية (Default) الخاصة بكل اختيار.

الاختيار	الشـرح
Title	وهــو عبــارة عــن أربعة سطور يتم كتابتها فوق كل صفحة من التقرير كمنوان لهذا التقرير.
Page Width	وهــو أكبر عدد من الحروف يمكن كتابته فى السطر الواحد . والقيمة

	المبدئية ٨٠ حرفا والمدى من ١ الى ٥٠٠ حرف.
Left Margin	وهـــو طول المسافة المحصورة بين حرف الصفحة الأيسر وأول حرف مطبوع . والقيمة المبدئية ٨ والمدى من صفر حتى عرض الصفحة.
Lines per page	وهــو أكبــر عدد من السطور يمكن طباعته في الصفحة . والعدد المبدئي (Default) هــو ٥٨ سطـــرا . و المدى من ٢٠ الى ١٠٠ سطر .
Double space report	وهـــو يسمـــح بترك سطر خال بين كل سطرين متتاليين . والوضع المبـــدئـــي يكـــون (Single-spacing) أي بدون سطور خالية .
Page eject before printing	و هـــو يؤدي الى تحــريك الـورقة الى بداية الصــفحة التاليــة عند بدء الطباعة . والوضع المبدئي (Yes) .
Page eject after printing	و هو يــؤدي الى تحريك ورقة خالية فــي نهاية الطباعة . و الوضع المبدئي (No) .
Plain page	وهـــو يعنــي أن التقرير لا يحـتوى على ارقام الصــفحات والتـــاريخ على الصفحات . والوضـع المبـــدئي هو (No) أى كتــابة هــذه البيانــات . وعــند تغييره الى(Yes) لايتم كتابة هـذه البيانات .

٨ - ١ - ٢ تجميع أو تصنيف السجلات (Grouping)

وهو الاختيار الثاني في عمود الاختيارات(Menu Bar) ويتم عن طريقه تجميع أو تصنيف السجلات تبعا للحقل الفهرسي(Index Field) الى مجموعات رئيسية ومجموعات فرعية . حيث يتم وضع كل مجموعة من السجلات التي تشترك في الحقل الفهرسي مع بعضها مع تحديد عنوان لهذه المجموعة . فمثلا اذا كـان الحقـل الفهرسي هو حقل تاريخ الالتحاق (Dat_Ent) ، فيمكن أن تظهر بيانات الطلبة الذين التحقوا في كل عام في مجموعة منفصلة .

وهذا الاختيار لا يتم اختياره الا فى الحالات التى تتطلب ذلك حيث يمكن للمستخدم عدم الدخول فى قائمة التجميع عند انشاء التقرير .

ويتم استخدام هذه القائمة باتباع الخطوات التالية :

١ - يتم نقل مؤشر عمود الاختيارات الى الاختيار (Groups) . فيتم فتح القائمة الخاصة به . انظر الشكل (٨- ٣)
٢ - يتم اختيار (Group On Expression). وهو يعنى تحديد الحقل المطلوب التجميع بناء عليه .
٣ - يتم الضغط على مفتاح (F10) لعرض أسماء الحقول واختيار اسم الحقل المطلوب ادخاله ثم الضغط على مفتاح الادخال .
٤ - يتم تحريك العمود الضوئى (Highlight) الى (Group Heading) وكتابة الاسم المطلوب لهذه المجموعة من السجلات ، ثم الضغط على مفتاح الادخال .
٥ - يتم الضغط على مفتاح (<--) للخروج من قائمة التجميع (Groups) . وفيما يلى جدول يوضح الاختيارات الخاصة بالتجميع وشرح كل منها .

```
Options        Groups        Columns        Locate        Exit   04:33:01 pm
               Group on expression
               Group heading
               Summary report only      No
               Page eject after group
               Sub-group on expression
               Sub-group heading

CREATE REPORT  ||<C:>||C:F.FRM        ||Opt: 1/9        ||      ||NumCap
                   Enter report title.  Exit - Ctrl-End.
```

شكل (٨ - ٣)

الاختيار	الشرح
Group on expression	ويتم عن طريقه كتابة اسم الحقل أو العلاقة التى

يتم التجميع أو التصنيف بناء عليها . حيث يمكن التجميع بناء على علاقة بين حقول معينة .

Group heading	ويتم عن طريقه اعطاء عنوان للمجموعة . ويمكن ادخال حتى ٤ سطور كعنوان .
Summary report only	وهو يؤدي الى طباعة معلومات مختصرة عن السجلات الموجودة في المجموعة .
Page eject-after group	وهو يؤدى الى طباعة كل مجموعة من السجلات في صفحة منفصلة .
Sub-group on-expression	وهو يؤدى الى انشاء مجموعات فرعية تبعا لحقل معين .
Sub_group heading	وهو يؤدي الى كتابة عنوان للمجموعة الفرعية عند الطباعة .

٨ - ١ - ٤ تخطيط الأعمدة (Column Layout)

والمقصود به تحديد مكان وعرض كل عمود يمثل حقلا معينا من حقول الملف . ولتنفيذ ذلك يتم تحديد مكان كل حقل والعنوان الخاص به ، وذلك كالآتى :

١- يتم تحريك مؤشر عمود القوائم (Menu Bar) الى الاختيار (Column) ، فيتم فتح القائمة الخاصة به . انظر الشكل (٨ - ٤)

```
                         Contents
                         Heading
                         Width                 0
                         Decimal places
                         Total this column
 Report Format
>>>>>>>>-------------------------------------------

CREATE REPORT  |<C:>|C:P.FRM              |Column: 1
   Position selection bar - ↑↓.  Select - ◄┘.  Prev/Next
     Enter a field or expression to display in the indicat
```

شكل (٨- ٤)

٢- يلاحظ وقوف العمود الضوئي (Highlight) على الاختيار (Contents) . ويمكن كتابة اسم الحقل المراد اختياره ، أو يتم الضغط على مفتاح (F10) لعرض قائمة الحقول والاختيار منها .

٣- يتم تحريك العمود الضوئي الى الاختيار (Heading) ثم كتابة العنوان المراد طباعته لهذا الحقل . ويمكن كتابة عنوان مختلف عن اسم الحقل لأنه لا يكون هناك شروط محددة لعدد حروف هذا العنوان . وذلك عكس اسم الحقل الذى يكون مقيدا بالشروط المعروفة . كما يمكن كتابة هذا العنوان على سطرين أو ثلاثة حسب الحاجة . كما يمكن ترك سطور خالية قبل هذا العنوان وذلك بالضغط على مفتاح الادخال قبل بدء الكتابة عددا من المرات يقابل عدد السطور المراد تركها خالية .

انظر الشكل (٨ - ٥)

```
Options     Groups     Columns     Locate        Exit
                        Contents               NAME
                        Heading                NAME
                        Width                   30
                        Decimal places
                        Total this column
 Report Format
>>>>>>>>NAME

        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

CREATE REPORT   <C:> C:F.FRM                 Column: 1
    Position selection bar - ↑↓.  Select - ←┘.  Prev/Next
       Enter up to four lines of text to display above the
```

شكل (٨ - ٥)

٤- يلاحظ ظهور العدد الممثل للعرض (Width) الذى سبق تحديده لهذا الحقل عند بداية انشاء ملف قاعدة البيانات .

٥- يلاحظ أيضا ظهور عنوان الحقل (Heading) وتحته علامات (X) بقدر عرض الحقل (Width) الذى سبق تحديده ، وذلك فى المستطيل الموجود أسفل الشاشة ، والذى يستخدم فى تحديد شكل التقرير .

٦- يتم الضغط على مفتاح (PgDn) لتوصيف عمود آخر فى التقرير. فيلاحظ ظهور قائمة خالية يتم من خلالها تحديد الحقل الثانى المطلوب تمثيله فى التقرير .

٧- يتم ادخال باقي الحقول المطلوب عرضها في التقرير بنفس الطريقة .
٨- يتم الضغط على مفتاح السهم يمين (<--) للخروج من قائمة الأعمدة (Columns) . ويلاحظ ظهور الشكل النهائي للتقرير على الشاشة .

ملاحظات

١- اذا كان الحقل حرفيا تظهر الحروف (X) بعرض الحقل الذى سبق تحديده . واذا كان الحقل عدديا تظهر الأعداد (٩) بعرض الحقل ايضا .
٢- عند زيادة عرض عنوان الحقل (Heading) عن عرض الحقل (Width) فان البرنامج يقوم بضبط عرض العمود حتى يغطى عرض عنوان الحقل .
٣- يلاحظ وجود الاختيار (Total this column) فى قائمة الأعمدة (Columns) . فاذا أريد تجميع الأعداد الموجودة فى هذا الحقل يتم تعديل الوضع المبدئى لهذا الاختيار من (No) الى (Yes) .

٨ - ١ - ٥ اختيار الحقول قبل تخزين الملف

يمكن الرجوع الى أي حقل وتعديل بياناته عن طريق قائمة (Locate) . ولتنفيذ ذلك يتم اتباع الخطوات الآتية :

١- يتم فتح قائمة (Locate) ويلاحظ ظهور أسماء الحقول الموجودة بالتقرير .
٢- يتم تحريك المؤشر الضوئي (Highlight) لاختيار الحقل المطلوب اختباره ، والضغط على مفتاح الادخال .
٣- يلاحظ ظهور القائمة الخاصة بهذا الحقل متضمنة اسم الحقل وعنوانه في التقرير وعرضه ويمكن تعديل هذه البيانات للوصول الى شكل التقرير المطلوب .

٨ - ١ - ٦ تخزين وتعديل التقرير

لتخزين التقرير يتم تحريك مؤشر عمود القوائم (Menu Bar) الى قائمة الخروج (Exit) ثم اختيار الأمر (SAVE) .

ولتعديل التقرير يتم اختيار قائمة التعديل (Modify) من القوائم الرئيسية الثمانية التى سبق ذكرها . ويتم اختيار التقرير (Report) من

هذه القائمة ، فيلاحظ ظهور قائمة بأسماء ملفات التقارير التي سبق تخزينها. فيتم اخـتيار التقرير المطلوب تعديله . ويلاحظ ظهور نفس القوائم المستخدمة في انشاء التقرير .

٨ - ١ - ٧ طباعة التقرير

يتم استخدام ملف التقارير في عرض بيانات سجلات محددة على الشاشة أو طباعتها على الطابعة . ولتنفيذ ذلك يتم اتباع الخطوات التالية :

١- يتم التأكد أولا من فتح ملف قاعدة البيانات المطلوب . كما يتم فتح ملف الفهرس (Index File) .
٢- يتم فتح قائمة الاسترجاع(Retrieve) واختيار (Report) .
٣- يتم اختيار وحدة الأقراص التي تحتوى على القرص المخزن به ملف التقارير المطلوب ، ثم اختيار الملف المطلوب .
٤- يتم تحديد شروط البحث(Search Conditions) ، ومجال البحث (Search Scope) ، لاختيار سجلات محددة حسب الحاجة .
٥- يتم اختيار(Execute the command) ، فيظهر السؤال التالي :

Direct the output to the printer? (Y/N)

٦- يتم التأكد من أن الطابعة جاهزة . ثم كتابة(Y) فيتم طباعة التقرير .
٧- اذا أريد عرض التقرير على الشاشة فقط يتم كتابة (N) أمام السؤال السابق .

٨ - ٢ انشاء العناوين المختصرة (Labels)

في بعض الأحيان يكون مطلوبا طباعة أو عرض عناوين مختصرة وسريعة (Labels) . وهذه العناوين تتضمن بعض البيانات الضرورية مثل الاسم والعنوان ورقم التليفون مثلا . وتتبع في انشاء ملف العناوين وتعديله نفس الخطوات السابق شرحها في انشاء وتعديل التقرير .

ولانشاء ملف العناوين المختصرة(Labels) ، يتم اتباع الخطوات التالية :

١- يتم فتح قائمة التجهيز(Set Up) واختيار ملف قاعدة البيانات (Data Base File) .
٢- يتم اختيار ملف الفهرس المطلوب فتحه .
٣- يتم فتح قائمة الانشاء(Create) ، واختيار (Label) .

٤- يتم اختيار وحدة الأقراص التي تحتوى على القرص المطلوب تخزين الملف عليه .
٥- يتم كتابة اسم ملف العناوين المطلوب انشاؤه والضغط على مفتاح الادخال .
٦- يلاحظ فتح قائمة العناوين مع وقوف المؤشر العلوى على قائمة(Options)، وبالتالى يتم فتح القائمة الفرعية الخاصة بها . و هذه القائمة تحتوى على اختيارات يتم عن طريقها تحديد أبعاد الطباعة .

٨ - ٢ - ١ تحديد أبعاد الصورة المطبوعة

لتحديد أبعاد الطباعة يتيح البرنامج ثلاثة أبعاد قياسية (Standard) ويتم ذلك باتباع الخطوات التالية :

١- مع وجود العمود الضوئي(Highlight) على(Predefined Size) يتم الضغط على مفتاح الادخال فيلاحظ ظهور أرقام تمثل الأبعاد القياسية للصورة المطبوعة مثل (3 1/2 X 15/16) by 3)) . ومع كل ضغطة على مفتاح الادخال تظهر أبعاد قياسية جديدة . والرقم السابق يؤدى الى طباعة تقرير أبعاده(٣' ١/٢ في ١٥/١٦ بوصة) ، وطباعة ثلاثة تقارير فى الصفحة الواحدة . أنظر الشكل(٨ - ٦)

```
Options                     Contents                          Exit
 Predefined size:        3 1/2 x 15/16 by 1

 Label width:            35
 Label height:           5
 Left margin:            0
 Lines between labels:   1
 Spaces between labels:  0
 Labels across page:     1

CURSOR:  <-- -->  | Delete char: Del | Insert row:    ^N | Insert:   Ins
 Char:    ← →     | Delete word:  ^T | Toggle menu:   F1 | Zoom in:  ^PgDn
 Word:  Home End  | Delete row:   ^U | Abandon:      Esc | Zoom out: ^PgUp
CREATE LABEL  |C:>|C:N.LBL          |Opt: 1/7          |        | Num
   Position selection bar - ↑↓.  Select - ←┘.  Leave menu - ↔.
   Select a standard label size: (Width x Height by Number across).
```

شكل (٨ - ٦)

٢- يمكن تعديل باقى الاختيارات الخاصة بأبعاد التقرير المختصر، كما يمكن

تترك القيم المبدئية(Default) الموجودة في الجدول كما هي حيث أنها تعتبر مناسبة .

والجدول التالى يوضح اختيارات هذه القائمة والقيم المبدئية (Default) لها .

الاختيار	الشرح
Label width	وهو أكبر عدد من الحروف في السطر الواحد من التقرير . والمدى من ١ الى ١٢٠ حرفا.
Label Hight	وهو يمثل عدد السطور في التقرير الواحد . والمدى من ١ الى ١٦ سطرا.
Left margin	وهو يمثل المسافة بين الحرف الأيسر للورقة وأول حرف مطبوع . والمدى من صفر الى ٢٥٠ حرفا.
Lines between labels	وهو يمثل المسافة الرأسية بالسطور بين سطور التقرير والمدى من صفر الى ١٦ سطرا .
Spaces between labels	وهو يمثل المسافة الأفقية بين التقارير والمدى من صفر الى ١٢٠ حرفا.
Labels across page	وهو يمثل عدد التقارير المطبوعة في الصفحة . والمدى من ١ الى ١٥ تقريرا .

ملاحظة

هذه القيم المبدئية تتغير بتغير الأبعاد القياسية للتقرير (Predefined Sizes).

٨ - ٢ - ٢ ادخال محتويات التقرير

يتم ادخال محتويات تقرير العناوين المختصرة(Label) عن طريق الخطوات التالية :

١- يتم تحريك المؤشر العلوى الى الاختيار (Contents) . فيتم فتح القائمة الخاصة به .

٢- يتم تحريك العمود الضوئى الخاص بهذه القائمة الى السطر الثانى والضغط على مفتاح الادخال . فيلاحظ ظهور العلامة < وظهور مؤشر صغير على هذا السطر يتم عن طريقه كتابة اسم الحقل المطلوب ادخاله .
٣- يمكن كتابة أسماء الحقول أو الضغط على مفتاح (F10) لاظهار القائمة الخاصة بها واختيار الحقل المطلوب .
٤- يمكـــن ادخـــال أكثر من حقل فى نفس السطر عن طريق كتابة العلامة (,) بين أسماء الحقول .

ملاحظة

استخدام علامة (,) بين أسماء الحقول يؤدى الى التخلص من المسافات الزائدة فى نهاية الحقل (Trimming) . أما اذا أريد الاحتفاظ بهـــذه المسافات بين الحقول فتستخدم علامة الجمع (+) بدلا من الفاصلة (,) .

٥- يتم الضغط على مفتـــاح الادخال ، فيتم ادخال هذا السطر والانتقال الى السطر التالى .
٦- يتم ادخال باقى السطور بنفس الطريقة .
٧- لتخزين تقرير العناوين المختصرة (Label) يتم تحريك المؤشر العلوى الى آخر اختيار وهو (Exit) ، ثم اختيار (Save) .

٨ - ٢ - ٣ طباعة تقارير العناوين المختصرة

لطباعة تقارير العناوين المختصرة لسجل معين أو لمجموعة من السجلات يتم اتباع الخطوات التالية :

١- يتم فتح قائمة الاسترجاع (Retrieve) واختيار (Label) .
٢- يتم تحديد وحدة الأقراص التى تحتوى على القرص المخزن عليه ملف التقارير المختصرة الذى سبق انشاؤه .
٣- يتم تحديد شروط البحث ومدى البحث كما سبق الايضاح .
٤- يتم اختيار (Execute the command) . فيظهر السؤال الآتى على الشاشة :

```
Direct the output to the printer ? (Y/N)
```

٥- يتم التأكد من توصيل الطابعة وتشغيلها ثم كتابة (Y) .

٨ - ٢ تلخيص البيانات (Summarizing Data)

توجد ثلاثة اختيارات فى قائمة الاسترجاع (Retrieve) تؤدى الى تجميع البيانات العددية في السجلات التى يتم اختيارها . وهذه الاختيارات هى (Sum) ، (Average) ، (Count) .

والاختياران (Sum) ، (Average) يعملان على الحقول العددية فقط . حيث يؤدى الاختيار (Sum) الى تجميع الحقول العددية فى الملف . كما يؤدى الاختيار (Average) الى حساب المتوسطات العددية للحقول العددية في الملف . والاختيار (Count) يحسب عدد السجلات التى تحقق شرطا أو شروطا معينة .

أنظر الشكل (٨ - ٧)

```
Set Up Create Update Position Retrieve Organize ModifyTools
                              List
                              Display
                              Report
                              Label
                              Sum
                              Average
                              Count
ASSIST          |<C:>||M                 |Rec: 1/4
Move selection bar - ↑↓. Select - ◄┘. Leave menu - ↔. He
           Display the totals of the specified numeric f
```

شكل (٨ - ٧)

الباب التاسع

ربط قواعد البيانات

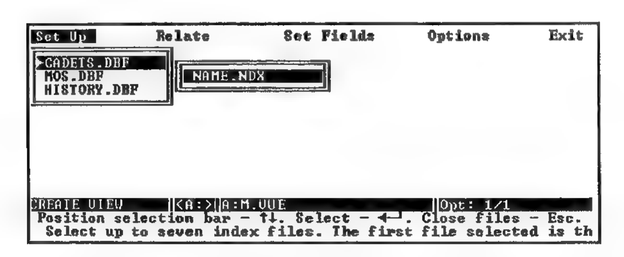
Relating Databases

عندما تكون قاعدة البيانات كبيرة ، أي تحتوى على عدد كبير من الحقول وعدد كبير من السجلات . فالأفضل في هذه الحالة انشاء عدة ملفات بدلا من ملف واحد. وذلك لأن الملف الكبير له عيوب كثيرة مثل الآتى :

١ - عند البحث عن بيان خلال الملف يستغرق البحث وقتا طويلا.
٢ - يحتل الملف جزءا كبيرا من الذاكرة المؤقتة عند تحميله . وهذا يؤثر على سرعة تشغيل البيانات .

وعند تقسيم قاعدة البيانات على عدة ملفات ، يجب أن تكون كل الملفات محتوية على حقل مشترك ، وهذا الحقل المشترك يجب أن يكون منفردا (Unique) . وعن طريق هذا الحقل يمكن ربط الملفات ببعضها. وهذا يتيح للمستخدم الاسترجاع السريع لأى سجل وتعديل البيانات المطلوب تعديلها. كما أن أى تعديل في حقل معين في أى ملف يؤثر في أى حقول معتمدة على هذا الحقل في الملفات الأخرى .

ويمكن أيضا استخدام ملف المنظر (View File). وهذا الملف عبارة عن ملف يتم فيه تخزين حقول من عدة ملفات مرتبطة ببعضها بواسطة حقل مشترك . ويمكن استرجاع هذا الملف في أى وقت واسترجاع البيانات المطلوبة والتى يقوم هذا الملف بتجميعها من الملفات المرتبطة به . انظر الشكل (٩- ١)



شكل (٩ - ١)

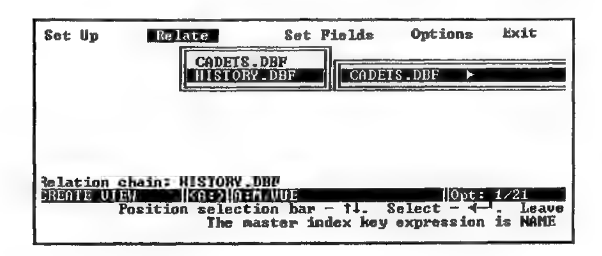
ملاحظة

لايستخدم ملف المنظر (view File) في اضافة (Append) سجلات جديدة ، ولكن يتم اضافة السجلات الجديدة عن طريق الملفات الأصلية .

٩ - ١ انشاء ملف المنظر (View File)

لانشاء ملف المنظر يتم أولا تحديد الملفات المطلوب ادخالها فيه . كما يتم تحديد الحقول المطلوبة من كـــل ملف والمطلوب ادخـــالها فـــى ملف المنظر (View File) . ولانشاء هذا الملف تتبع الخطوات التالية :

١ - يتم فتح قائمة الانشاء (Create) واختيار (View) .
٢ - يتم اختيار وحدة الأقراص التي يراد تخزين الملف فيها.
٣ - يتم كتابة اسم ملف المنظر المطلوب انشاؤه . ويـلاحظ ظهـــور عمـــود القـــوائم (Menu Bar) مع وقوف المؤشر الخاص به على الاختيـار (Set Up) . وبالتالى يتم فتح القائمة الخاصة به ، والتى تحتوي على ملفات قواعد البيانات الموجودة على القرص . فيتم اختيار الملف الأول وكذلك اختيار ملف الفهرس الخاص به (Index File) ثم الضغط على مفتاح السهم يمين (-->) للرجوع الى قائمة ملفات قواعد البيانات واختيار الملف الثانى وهكذا .
٤ - يتم تحريك المؤشر العلوى الى الاختيار (Relate) فيلاحظ فتح القائمة الخاصة بهذا الاختيار والتى تحتوى على أسماء الملفات التى تم اختيارها لادخالها فى ملف المنظر . أنظر الشكل (٩- ٢)



شكل (٩ - ٢)

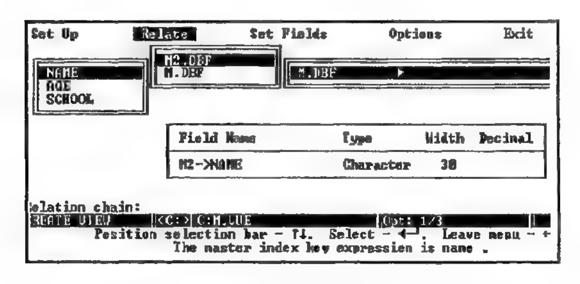
٥ - مع وجود العمود الضوئى (Highlight) على أول ملف يتم الضغط على مفتاح الادخال فيلاحظ ظهور قائمة الملفات المرتبطة بهذا الملف.
٦ - يتم اختيار كل ملف من هذه الملفات والضغط على مفتاح (F10) لاظهار قائمة بحقول هذا الملف. ومن هذه القائمة يتـم اختيار الحقل المراد استخدامه فى ربط هذا الملف بالملف الأول .

٧ - يعتبر الملف الأول هو الأصل وباقي الملفات مرتبطة بهذا الملف طبقا للحقل الفهرسي . أي أن السجلات في كل ملف يتم ترتيبها حسب ترتيب الحقل الفهرسي في الملف الأول .

٩ - ٢ اختيار حقول ملف المنظر (View File)

بعد ربط الملفات يتم تحديد الحقول المراد اختيارها من كل ملف لادخالها في ملف المنظر . ولتنفيذ ذلك تتبع الخطوات التالية :

١ - يتم تحريك المؤشر العلوي الى الاختيار (Set Fields) فيتم فتح القائمة الخاصة به . انظر الشكل (٩ - ٢)
٢ - يلاحظ ظهور قائمة بملفات قواعد البيانات التي تم ربطها بحيث يظهر الملف الأول في أول القائمة وبعده باقي الملفات .
٣- يتم تحريك المؤشر الضوئي على اسم كل ملف والضغط على مفتاح الادخال ، فتظهر قائمة بحقول هذا الملف ، ويتم اختيار الحقول المراد ادخالها في ملف المنظر . وهكذا .



شكل (٩- ٣)

٩ - ٣ تخزين ملف المنظر

لتخزين ملف المنظر (View File) يتم تحريك المؤشر العلوي الى الاختيار (Exit) والضغط على مفتاح الادخال . ويتم اختيار (Save) .

وعندما يراد تعديل الملف بعد ذلك يتم الدخول الى قائمة التعديل (Modify) من القائمة الرئيسية للبرنامج ، ثم اختيار(View) . ويلاحظ في هذه الحالة ظهور

نفس القوائم التي تظهر في حالة انشاء ملف منظر جديد .

٩ - ٤ فتح ملف المنظر

لفتح ملف المنظر يتم الدخول في قائمة التجهيز (Set Up) واختيار (View) وكتابة اسم الملف . ويتم استخدامه بعد ذلك في عرض وتعديل البيانات في الحقول التي تم اختيارها من الملفات المختلفة .

تحـــذير

يراعي عند اجراء تعديل في بيانات الحقول عدم تعديل بيانات الحقل المشترك الذي سبق تحديده .

٩ - ٥ استخدام الكتالوجات

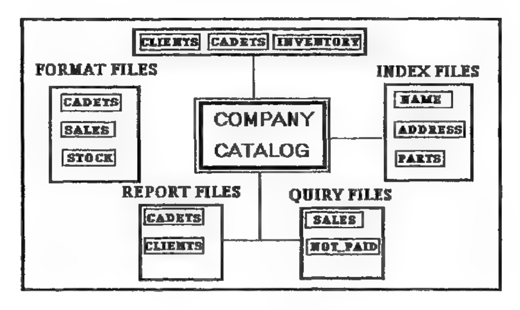
كما يستخدم نظام التشغيل (MS-DOS) نظام الفهارس (Directories) والفهارس الفرعية (Subdirectories) في تنظيم الملفات على القرص . فان برنامج (DBase III+) يستخدم الكتالوجات (Catalogs) في تنظيم الملفات الخاصة بقواعد البيـــانات (Database Files) والملـفات المرتبطة بها مثل ملفـات الفهـرس (Index Files) ومـلفـات البحث (Query Files) وملفات التقارير (Report Files) و... وهكذا .

انظر الشكل (٩- ٤)

وتفـيـد هـذه العمـلية عندمـا يكـون عـدد الملفات المخزنـة على القرص الصلب (Hard Disk) كبيرا جدا حيث يمكن ادخال كل مجموعة من قواعد البيانات التي تؤدي وظائف متقاربة في كتالوج منفصل . وعندما يراد استخدام الملفات الخاصة بقاعدة البيانات مثل ملفات التقارير أو ملفات البحث ، أو .الخ. فان قائمة الملفات التي تظهر للاختيار منها لا تحتوى الا على الملفات الموجودة في هذا الكتالوج ، وبالتالي لايتم عرض قوائم كبيرة للملفات .

وعند فتح الكتالوج ، فان أي ملفات جديدة يتم انشاؤها تضاف الى هذا الكتالوج ، وانشاء الكتالوج لايتم من خلال برنامج المساعد (Assistant) ولكن يتم ذلك من خلال أوامر النقطة (Dot Commands) التي سيتم شرحها فيما بعد .

أما فتح الكتالوج فيمكن أن يتم من خلال برنامج المساعد (Assistant) عن طريق قائمة التجهيز (Set Up) .



شكل (٩- ٤)

وعندما يراد تغيير الكتالوج الجارى استخدامه (Active Catalog) بكتالوج آخر يتم الرجوع الى قائمة التجهيز (Set Up) واختيار الكتالوج المطلوب . وفى هذه الحالة يصبح الكتالوج القديم غير مستخدم (Inactive) ويصبح الكتالوج الجديد هو الكتالوج المستخدم .

الباب العاشر

أوامر النقطة

Dot Commands

فى الأجزاء السابقة تم شرح استخدام برنامج المساعد (Assistant) فى انشاء ملفات قواعد البيانات وفتحها وكذلك انشاء باقى الملفات المرتبطة بها مثل ملف الفهرس (Index) والتقارير (Reports) والبحث (Query) والمنظر (View) و ... الخ . وكما كان واضحا من خلال هذا الشرح فان هذه العمليات يتم تنفيذها مـــن خلال قوائم تظهر على الشاشة .

وهناك طريقة أخرى لاتعتمد على القوائم ، ولكنها تعتمد على كتابة الأوامر مباشرة عند النقطة التى تظهر على الشاشة فوق عمود الحالة (Status Bar) . وهذه النقطة تظهر عند الخروج من القائمة الرئيسية باستخدام مفتاح الهروب (Esc) .

أنظر الشكل (١٠ - ١)

```
Command Line    ||<C:>||M                         ||Rec: 1/4
                         Enter a dBASE III PLUS command.
```

شكل (١٠ - ١)

ويلاحظ أن عمود الحالة (Status Bar) يبين هذا التحول من المساعد (Assistant) الى أمر (Command) .

ملاحظة

يمكن عن طريق مسح سطر من ملف المواصفات (Config.sys) جعل برنامج (DBase III +) يبدأ من أوامر النقطة (Dot Commands) بدلا من المساعد (Assistant) الذى يظهر عند بداية تشغيل البرنامج . وهذا السطر يكون كالآتى :

COMMAND = ASSIST

ويتم من خلال أوامر النقطة (Dot Commands) اجراء كل العمليات التى سبق شرحها وأداء عمليات اضافية أيضا . وهذه الطريقة فى ادخال الأوامر يتم استخدامها عادة بعد استخدام برنامج المساعد (Assistant) مدة كافية والتعود على أوامر البرنامج . ويجب ملاحظة أن كل أمر يتم ادخاله بواسطة برنامج المســاعد يظهـــر فوق عمود الحالة (Status Bar) فى نفس الوقت . وهكذا يمكن عن طريق برنامج المساعد (Assistant) معرفة شكل الأمر القابل (Syntax) عند اجراء أى عملية .

كما أن دراسة أوامر النقطة (Dot Commands) تعتبر أساسية لمن يريد كتابة البرامج بواسطة برنامج (+ DBase III) وبرامج عائلة (DBase) الأخرى .

١٠ - ١ ادخال الأوامر

يتم ادخال الأمر بعد النقطة (Dot) مباشرة . ثم يتم اضافة أي معاملات (Parameters) مطلوبة لهذا الأمر . والمعامل (Parameter) هو قيمة تساعد على تحديد عمل الأمر . ويمكن ادخال الأوامر بالحروف الصغيرة أو الكبيرة .

ويمكن تصحيح حروف الأمر باستخدام مفاتيح التصحيح المعتادة مثل مفتاح (Backspace) لحذف الحروف ، ومفاتيح السهم يمين والسهم يسار (<--- , --->) لتحريك المؤشر الى مكان الحرف وتصحيحه . وبعد الانتهاء من كتابة الأمر يتم ادخاله بالضغط على مفتاح الادخال .

ملاحظة

اذا حدث خطأ في كتابة الأمر تظهر الرسالة التالية :

Do you want some help ? (Y/N)

وعند كتابة (Y) تظهر قائمة المساعدة (Help) .
وعند كتابة (N) تظهر النقطة مرة ثانية لاعادة ادخال الأمر من جديد .

وهناك بعض الأوامر التي لاتحتاج الى معاملات (Parameters) مثل الأمر (HELP) والأمر (ASSIST) والأمر (SET) . فالأمر (HELP) يؤدي الى ظهور شاشات المساعدة (Help Screens) التي من خلالها يمكن التعرف على شكل كل أمر (Syntax) وخصائصه .

أما الأمر (ASSIST) فانه يؤدي الى ظهور القوائم الخاصة ببرنامج المساعد (Assistant) واجراء العمليات المطلوب تنفيذها من خلال هذا البرنامج .

والأمر (SET) يمكن استخدامه منفردا بدون معاملات . وهذا يؤدي الى ظهور قائمة يمكن من خلالها تغيير مواصفات البرنامج (Configuration) الخاصة بالشاشة ولوحة المفاتيح ... الخ . وعند اتباعها بمعاملات أخرى فانها تؤدي عملا محددا يتوقف على هذه المعاملات ، مثل (SET STATUS ON) ، (SET INDEX TO) و ... الخ .

ملاحظة

يمكن أيضا استخدام برنامج المساعد (Assistant) بالضغط على مفتاح (F2).

١٠ - ٢ عرض التاريخ (Display History)

يقوم برنامج (+ DBase III) بتخزين آخر أوامر تم ادخالها في مخزن مؤقت (Buffer). وهذا المخزن يخزن حتى ٢٠ أمرا . وعندما يراد عرض آخر أوامر تم ادخالها يتم الضغط على مفتاح السهم لأعلى (↑) . وفي كل مرة يتم الضغط على هذا المفتاح يظهر آخر أمر تم ادخاله .

ويفيد ذلك عندما يراد ادخال بعض الأوامر عدة مرات ، فيكفي في هذه الحالة الضغط على مفتاح الاتجاه لأعلى (↑) عدة مرات للوصول الى الأمر المطلوب ادخاله والضغط على مفتاح الادخال . وهذا يوفر الوقت اللازم لكتابة الأمر من جديد .

١٠-٣ تنفيذ عمليات قاعدة البيانات بواسطة الأوامر

كما سبق الايضاح فان جميع العمليات السابق شرحها ، والتى يتم تنفيذها من خلال برنامج المساعد (Assistant) ، يمكن تنفيذها من خلال الأوامر (Commands). وذلك بالاضافة الى بعض العمليات الأخرى التى يتم تنفيذها بواسطة الأوامر فقط .

وكتابة الأوامر عادة تكون أسرع من استخدام القوائم خاصة عند استخدام مفتاح السهم لأعلى (↑) لاستخدام الأوامر التى سبق ادخالها . وفي الأجزاء التالية يتم توضيح بعض العمليات التي سبق شرح تنفيذها من خلال برنامج المساعد (Assistant) مع شرح كيفية استخدام أوامر النقطة (Dot Commands) فى تنفيذها .

١٠ - ٣ - ١ انشاء واستخدام الكتالوجات

يمكن تجميع كل مجموعة من اللفات فى كتالوج منفصل كما سبق الايضاح . ولانشاء كتالوج جديد اسمه (Mycat) مثلا يتم تنفيذ الآتى :

١ - يتم كتابة الأمر التالى بعد النقطة (Dot) مباشرة :

```
SET CATALOG TO Mycat
```

يلاحظ ظهور السؤال التالي على الشاشة :

Create a new catalog File ? (Y/N)

٢ - يتم كتابة (Y) فيتم انشاء ملف الكتالوج بالاسم (Mycat) .

ملاحـــظة

بعد انشاء الكتالوج ، فان أي ملفات يتم انشاؤها أو استخدامها تضاف الى هذا الكتالوج . ويمكن اغلاق هذا الكتالوج باستخدام الأمر

SET CATALOG OFF

كما يمكن فتح كتالوج آخر باستخدام الأمر

SET CATALOG TO

ثم كتابة اسم الكتالوج المطلوب .

١٠ - ٣ - ٢ انشاء ملف قاعدة البيانات

لانشاء ملف قاعدة البيانات يستخدم الأمر (CREATE) مع اسم الملف المراد انشاؤه . فمثلا عندما يراد انشاء ملف اسمه (Myfile) يتم تنفيذ الآتي :

١ - يتم كتابة الأمر التالي بعد النقطة (Dot) مباشرة

CREATE Myfile

٢ - يلاحظ ظهور الشاشة المبينة في الشكل (١٠ - ٢) والتي عن طريقها يتـــم ادخال أسماء الحقـــول والبيـــانات الخاصة بعرض الحقل ونوعه و ... الخ .
٣ - يتم الضغط على مفتاحي (Ctrl-End) لتخزين الملف .
٤ - يتم الضغط على مفتاح الادخال لتأكيد الرغبة في تخزين الملف .
٥ - يلاحظ ظهور السؤال الآتي على الشاشة .

Input data records now? (Y/N)

٦ - يتم كتابة (N) عند عدم الرغبة في ادخال بيانات الملف في هذا

الوقت . واذا أريد ادخال البيانات يتم كتابة (Y) فتظهر شاشة ادخال البيانات كما سبق الايضاح .

```
                                                          Bytes remaining: 4000
 CURSOR  <-- -->   |   INSERT      |   DELETE      |  Up a field:      ↑
  Char:    ← →     |  Char:  Ins   |  Char:   Del  |  Down a field:    ↓
  Word: Home End   |  Field: ^N    |  Word:   ^Y   |  Exit/Save:     ^End
  Pan:    ^← ^→    |  Help:  F1    |  Field:  ^U   |  Abort:          Esc
      Field Name  Type      Width  Dec       Field Name  Type      Width  Dec
  1               Character
CREATE            ||<C:>||FG                ||Field: 1/1            ||
                         Enter the field name.
Field names begin with a letter and may contain letters, digits and under
```

شكل (١٠ - ٢)

١٠ - ٢ - ٣ فتح ملف قاعدة البيانات

يتم فتح ملف قاعدة البيانات باستخدام الأمر (USE) . فاذا أريد مثلا فتح الملف الذى سبق انشاؤه (Myfile) يتم كتابة الأمر التالى بعد النقطة :

```
USE Myfile
```

واذا أريد رؤية أسماء الملفات واختيار الملف المطلوب فتحه يستخدم الأمر التالى :

```
USE ?
```

وفي هذه الحالة تظهر أسماء جميع ملفات قواعد البيانات الموجودة على القرص أو الفهرس الفرعى المستخدم .

١٠-٢-٤ تعديل تركيب ملف قاعدة البيانات (Structure)

لتعديل تركيب ملف قاعدة البيانات يستخدم الأمر التالى :

```
MODIFY STRUCTURE
```

ويجب أن يكون الملف قد سبق فتحه باستخدام الأمر (USE) كما سبق

الايضاح . وفـي هـذه الحالة تظهر الشاشـة الخاصـة بتوصيـف الحـقول المبينة فـي الشـكل (١٠ - ٢) .

١٠-٢-٥ انشاء ملفات شاشة الادخال (Format File)

كما سبق الايضاح، فان من المهم تصميم شاشات لادخال البيانات واضحة وسهلة بالنسبة للشخص القائم بعملية ادخال البيانات . ولانشاء شاشة ادخال البيانات يستخدم الأمر التالى :

```
Create Screen
```

ثم كتابة اسم الملف المطلوب انشاؤه .

يلاحظ فى هذه الحالة ظهور القوائم التى سبق شرحها فى الجزء الخاص بتصميم شاشات الادخال .

١٠ - ٢ - ٦ فتح ملفات شاشة الادخال

يتم فتح ملفات شاشة الادخال باستخدام الأمر :

```
SET FORMAT TO
```

ثم كتابة اسم الملف المراد فتحه .

ويمكن عرض أسماء ملفات شاشة الادخال الموجودة على القرص باستخدام الأمر :

```
SET FORMAT TO ?
```

وتظهر فى هذه الحالة قائمة بكل ملفات شاشة الادخال المخزنة على القرص .

ويمكن اغلاق ملف شاشة الادخال المفتوح باستخدام الأمر التالى :

```
SET FORMAT TO
```

دون كتابة أسماء أى ملفات بعده .

١٠ - ٢ - ٧ استخدام الأمر (BROWSE)

يستخدم الأمر (BROWSE) كما سبق الايضاح في عرض شاشة موضحا بها بيانات مجموعـة مـن السجلات على هيئة أعمدة تمثل الحقول بحيث يكـون كـل سجل في سطر . وعن طريق هذه الشاشة يمكن تعديل بيانات أي سجل أو اضافة سجلات جديدة . ولتنفيذ ذلك عن طريق أمر النقطة (Dot Command) يتم كتابة الأمر كالآتي :

```
BROWSE Fields
```

ثم كتابة أسماء الحقول المراد عرضها على الشاشة . فاذا أريد مثلا عرض حقول الاسم (Name) ، والعنوان (Address) ، والوظيفة (Job) ، يتم كتابة الأمر التالي :

```
BROWSE fields name , address , job
```

أنظر الشكل (١٠ - ٢) .

```
CURSOR   <-- -->    |        UP     DOWN  |    DELETE      | Insert Mode:   Ins
  Char:     ←   →   | Record:  ↑     ↓    | Char:    Del   | Exit:          ^End
  Field: Home  End  | Page:  PgUp   PgDn  | Field:   ^Y    | Abort:          Esc
  Pan:     ^←  ^→   | Help:   F1          | Record:  ^U    | Set Options: ^Home

NAME--------------------------  ADDRESS---------------------  PHONE-----
mohamed hasan fathy             12-ain shams                  36526756
ahmed soliman tarek             40-ahram street               6789889
WALAA MOSTAFA                   NASR CITY                     6394588
HAYTHAM MOSTAFA                 NASR CITY                     7428953

                         View and edit fields.
```

شكل (١٠ - ٢)

١٠ - ٢ - ٨ استخدام الأمر (GOTO)

عـن طريق الأمر (GOTO) يمكن تحريك مؤشر البرنامج الى سجل محدد ، وذلك حتى يمكن تعديل بيانات سجل معين أو عرضها حسب الحاجة . ولاستخدام الأمر (GOTO) في الوصول الى السجل رقم ٣ مثلا يتم كتابة الأمر التالي :

```
GOTO 3
```

١٠ - ٢ - ٩ استخدام الأمر (EDIT)

يستخدم الأمر (EDIT) عادة بعد وضع المؤشر على سجل معين ثم كتابة الأمر في أبسط صورة له كالآتى :-

```
EDIT
```

وفى هذه الحالة تظهر شاشة ادخال البيانات التى يتم عن طريقها تعديل البيانات المطلوبة .

١٠ - ٢ - ١٠ استخدام الأمر (APPEND)

يستخدم الأمر (APPEND) فى اضافة سجل بعد آخر سجل به بيانات فى الملف . ويكتب الأمر كالآتى :

```
APPEND
```

فتظهر شاشة ادخال خالية ، يتم عن طريقها اضافة السجل الجديد .

١٠-٢-١١ انشاء واستخدام ملف الفهرس (Index File)

يتم انشاء ملف الفهرس باستخدام الأمر (INDEX ON) . فمثلا عندما يراد انشاء ملف فهرس بناء على حقل الاسم (Name) مع تسمية هذا الملف (Name) يتم كتابة الأمر كالآتى :

```
INDEX ON Name TO Name
```

مع ملاحظ أن برنامج (+ DBase III) يضيف لملف الفهرس الامتداد (.NDX) .

ويتم فتح ملف الفهرس بطريقتين . الطريقة الأولى عند فتح ملف قاعدة البيانات باستخدام الأمر التالى :

```
USE Myfile INDEX Name
```

حيث (Myfile) هو اسم ملف قاعدة البيانات
و (Name) هو اسم الملف الفهرسى الخاص بهذا الملف .

والطريقة الثانية باستخدام الأمر التالى

```
SET INDEX TO Name
```

١٠ - ٢ - ١٢ انشاء واستخدام ملف الفرز (Sorting)

كما سبق الايضاح فان الفرز يؤدى الى انشاء ملف جديد مرتب بالترتيب المطلوب . ولتنفيذ هذه العملية باستخدام أوامر النقطة (Dot Commands) ، يتم كتابة الأمر التالى :

```
SORT TO Sname ON Name
```

حيث Sname هو اسم الملف الجديد المطلوب انشاؤه .
Name هو اسم الحقل الذى يتم الترتيب بناء عليه .

وفى هذه الحالة يتم انشاء ملف جديد اسمه (Sname) مرتب حسب الترتيب الهجائى لحروف حقل الاسم . مع ملاحظة أن الترتيب يكون تصاعديا (Ascending) من الأقل فالأكبر . فاذا أريد عكس هذا الترتيب يتم اضافة الحرف (D) فى نهاية الأمر كالآتى :

```
SORT TO Sname ON Name /D
```

ملاحظة

ماسبق ذكره فى هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، FoxBase + ، FoxPro

الباب الحادى عشر

كتابة البرامج

من الامكانيات المتقدمة لبرنامج (+ DBase III) وكذلك باقي برامج عائلة (DBase) مثل (DBase IV) ، (FoxBase +) ، (FoxPro) أنها تستخدم كأداة برمجة قوية (Programming Tool) . وهو ما يميزها عن كثير من اللغات الأخرى لسهولة كتابة البرامج بها واستخدام كثير من الدوال المبنية فيها (Built in Functions) .

وكتابة البرامج تعتمد على تجميع أوامر النقطة (Dot Commands) في ملف مع اضافة بعض أوامر التحكم التى تساعد على التحكم فى تسلسل تنفيذ الأوامر . ويتم تشغيل هذه الأوامر عن طريق تشغيل الملف الذى يحتوى عليها والذى يسمى ملف البرنامج (Program File) .

ويمكن دراسة المثال التالى لتوضيح أهمية كتابة البرامج بواسطة (DBase III+) .

مثال

نفرض أنه يوجد ملف قاعدة بيانات يسمى (Money.DBF) يحتوى على معلومات عن أشخاص مدينين ، ويراد معرفة بيانات الأشخاص الذين مازالوا مدينين وكذلك مجموع الديون المستحقة والحصول على تقرير بذلك . هذه العمليات يتم تنفيذها من خلال أوامر النقطة (Dot Commands) التالية :

```
USE MONEY INDEX NAME ↵
LIST FOR Owing ↵
CLEAR ↵
SUM Amount_due FOR Owing ↵
REPORT FORM MONEY FOR Owing TO PRINT ↵
USE ↵
```

وهذه الأوامر يمكن توضيحها كالآتى :

١ - الأمر الأول يؤدى الى فتح الملف (Money.dbf) بالاضافة الى فتح الفهرس (Name.ndx) .
٢ - الأمر الثاني يؤدى الى عرض بيانات الأشخاص المدينين .
٣ - الأمر الثالث يؤدى الى مسح الشاشة .
٤ - الأمر الرابع يؤدى الى تجميع المبالغ للأشخاص المدينين .
٥ - الأمر الخامس يؤدى الى طباعة تقرير بيانات المدينين .
٦ - الأمر السادس يؤدى الى اغلاق جميع الملفات المفتوحة .
٧ - علامة (↵) تعنى الضغط على مفتاح الادخال بعد كل أمر .

ويمكن كتابة هذه الأوامر في ملف وتسمية هذا الملف (Owing.prg) ، ويمكن تنفيذ نفس العمليات السابقة بكتابة أمر واحد أمام النقطة (Dot) وهو كالآتى :

```
DO Owing
```

والضغط على مفتاح الادخال .

وهذا مثال بسيط ولكنه يوضح كيف يمكن توفير الوقت والجهد عن طريق كتابة البرامج من خلال (+ DBase III) وبرامج عائلة (DBase) الأخرى .

١١ - ١ أهمية كتابة البرامج

تسمح كتابة البرامج بواسطة برنامج(+ DBase III) أو أى برنامج من عائلة (DBase) بتسهيل تعامل المستخدم مع قواعد البيانات . وذلك عن طريق عرض القوائم الواضحة التى يستطيع من خلالها تنفيذ عمليات برامج ادارة قواعد البيانات مثل ادخال سجلات جديدة أو تعديل بيانات السجلات أو تحديث البيانات المخزنة أو عرض البيانات أو طباعتها وهكذا . كما تسمح كتابة البرامج أيضا بوضع وسائل التأمين للبيانات بحيث لايتعامل مع هذه البيانات الا الأشخاص المكلفون بذلك . كما أنها تتيح للمستخدم استخدام كل خواص برامج ادارة قواعد البيانات دون الحاجة لوجود خبرة سابقة بهذه البرامج أى أنها تمثل حلقة الاتصال بين المستخدم وبين قاعدة البيانات .

١١ - ٢ انشاء ملف البرنامج (Program File)

لانشاء ملف البرنامج يستخدم الأمر التالى :

```
MODIFY COMMAND
```

ثم اسم الملف المطلوب انشاؤه .

فاذا كان هذا الملف موجودا على القرص تظهر قائمة الأوامر الخاصة به على الشاشة . واذا لم يكن موجودا ، يتم فتح ملف جديد . مع ملاحظة ظهور قائمة مساعدة (Help Menu) أعلى الشاشة لتوضيح مفاتيح التصحيح المطلوب استخدامها أثناء كتابة البرنامج . ويمكن الغاء هذه القائمة بالضغط على مفتاح (F1) ، اذا أريد الاستفادة بالشاشة كلها فى كتابة البرنامج .

وعند الانتهاء من كتابة البرنامج يتم تخزينه باستخدام مفتاحى (Ctrl-End) أو مفتاحى (Ctrl-W) . وعند عدم الرغبة فى تخزين الملف أو التعديلات التى تم

ادخالها عليه ، يتم الضغط على مفتاح (Esc) فيظهر السؤال التالى على الشاشة :

```
Abort editing ? (Y/N)
```

وعند كتابة (Y) يعود البرنامج الى مشيرة النقطة (Dot Prompt) .

مـلاحظة

ما سبق ذكره فى هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل (DBase IV) ، (FoxBase +) ، (FoxPro) .

الباب الثانى عشر

خصائص كتابة البرامج

١٢ - ١ ماهو البرنـامج

البرنامج هو مجموعة من الارشادات والأوامر التي توجه الحاسب الى تنفيذ مهام أو وظائف معينة بتسلسل محدد . والبرنامج بصفة عامة لايختص بالحاسب فقط ، ولكن أى مجموعة من الأوامر والارشادات المرتبة بتسلسل معين هي برنامج . فمثلا الوصفة الخاصة بالطهى ، والتي تشمل خطوات محددة ومتسلسلة بحيث لايمكن مثلا طهى البطاطس قبل تقشيرها وتقطيعها ، هذه الوصفة تعتبر برنامج .

والفرق بين البرنامج الذى يعطي للانسان والبرنامج الذى يعطى للحاسب هو أن الأول يمكن للانسان تعديل بعض الأوامر فيه حسب اختياره بين البدائل المختلفة . أما بالنسبة للحاسب فانه يلتزم بالأوامر الموجودة ولاينفذ الا الأوامر التى تخضع لقواعد معينة سبق تحديدها .

لذلك فان كتابة البرامج للحاسب تتطلب كتابة جميع الأوامر والارشادات بتسلسل دقيق حتى تؤدى الوظيفة المطلوبة .

١٢ - ٢ لغة كتابة البرامج

لغة كتابة البرامج بواسطة (+ DBase III) أو برامج عائلة (DBase) الأخرى مثل (DBase IV) ، (+ FoxBase) ، (FoxPro) هي لغة سهلة وواضحة لأنها تحتوى على الكلمات الانجليزية الواضحة مثل (IF) ، (DO) ، ... الخ . ولكن الحاسب فى الواقع لايفهم هذه الكلمات ، لذلك فان المترجم (Interpreter) الخاص بالبرنامج يقوم بقراءة كل أمر يتم ادخاله وترجمته الى لغة الآلــة (Machine Language) التي يفهمها الحاسب . فمثلا عند ادخال الأمر (USE) ، فان الحاسب يقوم بمقارنة كل حرف فى الأمر (U-S-E) بقائمة الأوامر(Table of commands) المخزنة في الحاسب ، وهذه الخطوة تسمي (Parsing) أي تقسيم الأمر. وعندما يجد المترجم (Interpreter) تطابقا بين هذا الأمر الذى تم ادخاله ، وبين أحد الأوامر الموجودة في قائمة الأوامر ، فانه ينفذ التعليمات الخاصة بهذا الأمر. وعندما لايجد الأمر مطابقا فانه يعطى رسالة خطأ (Error message). وأخطاء كتابة البرنامج يمكن أن تحدث لأحد الأسباب الآتية :

١ - أخطاء في هجاء الأمر .
٢ - عدم فصل الأمر عن المعاملات (Parameters) بمسافة خالية (Space) على الأقل .
٣ - أخطاء القواعد (Syntax Errors) وتعنى عدم تطبيق القواعد الخاصة بهذا الأمر تطبيقا سليما .
٤ - أخطاء فى التسلسل المنطقى للأوامر.

١٢ - ٣ كتابة وتصحيح البرنامج

يحتوى برنامج (+ DBase III) على برنامج معالجة كلمات مبنى داخل البرنامج (Built in) يستخدم فى كتابة وتصحيح ملفات البرامج . ولتشغيل برنامج معالجة الكلمات يستخدم الأمر (MODIFY COMMAND) ثم كتابة اسم الملف المطلوب كتابته . والبرنامج يضيف الامتداد (.prg) آليا الى اسم الملف المطلوب انشاؤه . فمثلا عند كتابة برنامج يراد تسميته (Test.prg) يتم ادخال الأمر الآتى بعد مشيرة النقطة (Dot prompt).

```
MODIFY COMMAND Test
```

والضغط على مفتاح الادخال .

وعندما يراد تعديل ملف برنامج موجود ، يتم استخدام نفس الأمر (MODIFY COMMAND) وبعده اسم الملف . وعند فتح ملف البرنامج تظهر قائمة المساعدة (Help menu) أعلى الشاشة لتوجه مخطط البرامج الى المفاتيح المستخدمة فى تحريك المؤشر (Cursor) وأداء عمليات التصحيح المختلفة . ولاخفاء هذه القائمة يستخدم المفتاح (F1) . كما يمكن اعادتها مرة ثانية باستخدام نفس المفتاح . انظر الشكل (١٢ - ١) .

CURSOR <-- -->	INSERT	DELETE	
Char: ← →	Char: Ins	Char: Del	Up a field: ↑
Word: Home End	Field: ^N	Word: ^Y	Down a field: ↓
Pan: ^← ^→	Help: F1	Field: ^U	Exit/Save: ^End
			Abort: Esc

شكل (١٢ - ١)

ويمكن استخدام أى برنامج معالجة كلمات (Word processor) ، أو مصحح نصوص (Text editor) فى كتابة برامج (+ DBase III) على أن يكون من البرامج التى تسمح بانشاء ملفات آسكى (ASCII) . وعند استخدام أى

برنامج معالجة كلمات غير البرنامج المستخدم يجب التأكد أن الملف لايحتوى على أي شفرة تشكيل (Formating Code) مثل الحروف البارزة (Boldface) أو الخطوط السفلية (Underline) أو أي تشكيل آخر للصفحة .

ومن البرامج التي توفر كتابة نصوص بهذه الطريقة برنامج (Framework II) وبرنامج (Wordstar) الذى يسمح بتكوين ملفات غير وثائقية (Nondocument) .

وعند كتابة البرامج بواسطة أي برنامج معالجة كلمات يجب اضافة الامتداد (.prg) الى اسم البرنامج .

ملاحظة

اذا كانت الذاكرة المؤقتة (RAM) فى الحاسب كافية . يمكن كتابة اسم برنامج معالجة الكلمات المراد استخدامه داخل ملف المواصفات (Config.sys) بحيث يصبح هو المصحح المبدئى (Default) .

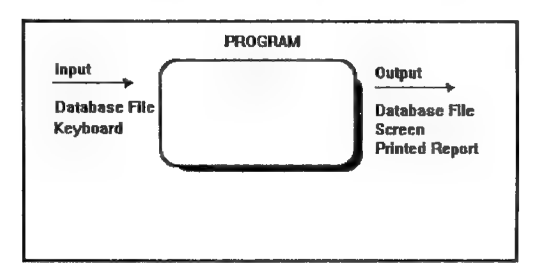
ويفضل استخدام برامج معالجة الكلمات عندما يكون ملف البرنامج المطلوب كتابته كبيرا . حيث أن المصحح الموجود فى برنامج (DBase III +) يكتب حتى ٥٠٠٠ حرف فقط . أى ما يقرب من ٢٠٠ سطر .

١٢ - ٤ تشغيل البرنامج

عندما يراد تشغيل البرنامج يتم كتابة الأمر (DO) ثم اسم ملف البرنامج الذى سبق انشاؤه . فيقوم برنامج (DBase III +) بفتح ملف البرنامج وقراءة كل سطر فيه بدءا من أول سطر مع تنفيذ كل أمر بالتسلسل . وعندما يصل الى آخر أمر وينفذه يعود الى مشيرة النقطة (Dot prompt) مرة ثانية .

١٢ - ٥ المدخلات والمخرجات (Input and Output)

أى برنامج تكون له مدخلات ومخرجات ، وهى التى تمثل مصدر المعلومات (Source) ، ومكان وصول هذه المعلومات (Destination) على الترتيب . فمثلا المدخلات يمكن أن تأتى من حقول قاعدة البيانات المستخدمة ، ويمكن أن يدخلها المستخدم عن طريق لوحة المفاتيح (Keyboard) أثناء تشغيل البرنامج . أما المخرجات فقد تكون قوائم بيانات السجلات على الشاشة أو التقارير المطبوعة فى الطابعة . انظر الشكل (١٢ - ٢) .



شكل (١٢ - ٢) المدخلات والمخرجات

والبرنامج الذي يتم كتابته يجب أن يوفر التحكم في المدخلات والمخرجات بأقل تدخل من المستخدم . حيث تنحصر وظيفة المستخدم في ادخال البيانات والحصول على المعلومات المطلوبة بعرضها على الشاشة أو طباعتها على الطابعة .

ويتم التحكم في المدخلات والمخرجات عن طريق الشاشات المجهزة (Customized Screens) . فبالنسبة للمدخلات تعمل هذه الشاشات كمحادثة مباشرة (Dialog) بين المستخدم والحاسب ، مع الرسائل التوضيحية التي تساعد على توجيه المستخدم أو تحذيره عند حدوث أي خطأ في ادخال البيانات . كما يمكن تصميم شاشات ادخال البيانات لتماثل النماذج المستخدمة في بعض النظم حتى يتمكن المستخدم من ادخال البيانات من النماذج الموجودة عنده بسهولة . وهذا يؤدي الى سهولة تشغيل البرنامج بواسطة أي شخص دون الحاجة الى أشخاص مؤهلين ذوي خبرة ببرامج ادارة قواعد البيانات . انظر الشكل (١٢ - ٣)

وبالنسبة للمخرجات يمكن أيضا استخدام الشاشات المجهزة (Customized Screens) في عرض البيانات التي يقوم المستخدم بادخالها ، واعطائه الفرصة للتأكد من صحتها (Validation) قبل تخزينها ، وكذلك اعطائه الفرصة لاسترجاع أي بيانات على الشاشة أو طباعتها على الطابعة .

```
Set Up              Modify              Options              Exit

                    _ CADETS INFORMATION

CADET NO  99999     RELATIVES   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HOBBIES   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX      DATE_SEC  99999
NO_BROTHER  99999          FATHER JOB               XXXXXXXXXXXXX

CREATE SCREEN  ||<A:>||A:CADETS.SCR          ||Pg 01 Row 03 Col 23||Ins
     Enter text.  Drag field or box under cursor with ←┘.  F10 for me
                  Screen field definition blackboard
```

شكل (١٢ - ٢) شاشة الادخال

وعن طريق البرنامج أيضا يمكن تخزين البيانات التي يقوم المستخدم بادخالها في أماكن مؤقتة في الذاكرة تسمى متغيرات الذاكرة (Memory Variables). ولايتم نقل هذه البيانات الى الحقول الخاصة بها في قاعدة البيانات الا بعد أن يتأكد المستخدم من صحة البيانات التي قام بادخالها . وهذه الطريقة تفيد في تحقيق البيانات (Validation) والتأكد من صحتها . حيث أن البيانات غير الصحيحة تؤدي دائما الى مخرجات غير صحيحة . وهو مايعرف في عالم الحاسب (Garbage in Garbage out) ويختصر (GIGO) . وهو يعنى أن جودة المخرجات ترتبط ارتباطا كبيرا بجودة المدخلات .

وعن طريق البرنامج أيضا يمكن تحقيق تكامل قاعدة البيانات (Database Integrity) ووحدة البيانات . فاذا كان هناك عدة مستخدمين يقومون بادخال البيانات ، يتم التأكد من دخول البيانات السليمة بواسطة الأشخاص المسموح لهم بادخال هذه البيانات. وتفيد متغيرات الذاكرة (Memory Variables) أيضا في تحقيق ذلك . وهذه المتغيرات سيتم شرحها بالتفصيل فيما بعد .

وعند كتابة البرنامج يجب دراسة كل الأخطاء المتوقعة من المستخدم وكتابة الأوامر التي تؤدى الى عدم توقف البرنامج بالاضافة الى تنبيه المستخدم الى الخطأ وطريقة اصلاحه . فمثلا عندما يكون هناك سؤال ينتهي بالاختيار بين (Yes) أو (No) يجب التأكد أن البرنامج يعرف ماذا يفعل عندما يدخل المستخدم أى حرف آخر غير (N) أو (Y) عن طريق الخطأ . وهذا سيتم ايضاحه فيما بعد بالتفصيل .

١٢ - ٦ التحكم فى البرنامج

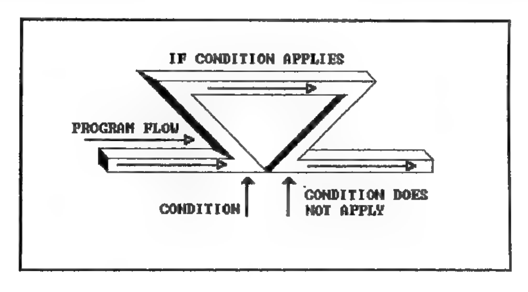
كما سبق الايضاح فان المترجم (Interpreter) يقوم بقراءة أوامر البرنامج وترجمتها بالتسلسل من البداية الى آخر أوامر البرنامج . ثم يقوم الحاسب بتنفيذ كل أمر فور ترجمته بنفس التسلسل .

ولكن فى بعض الأحيان يراد تنفيذ بعض الأوامر عدة مرات ، أو يراد الرجوع الى بعض الأوامر التى سبق تنفيذها ، أو يراد تخطى بعض الأوامر والذهاب الى أوامر فى مكان آخر من البرنامج . وفى جميع هذه الأحوال يراد تغيير تسلسل تنفيذ أوامر البرنامج .

ولتنفيذ ذلك تستخدم أوامر التحكم التى يؤدى بعضها الى تفرع البرنامج الى مكان آخر تفرعا غير مشروط . وبعضها يؤدى الى تفرع البرنامج تفرعا مشروطا يتوقف على تحقيق شرط معين أو عدم تحقيقه ، وبعضها يؤدى الى تكرار تنفيذ مجموعة من الأوامر بناء على شرط معين . وهذه الأوامر تعطى مخطط البرامج قدرة ومرونة عالية .

١٢ - ٦ - ١ التفرع المشروط

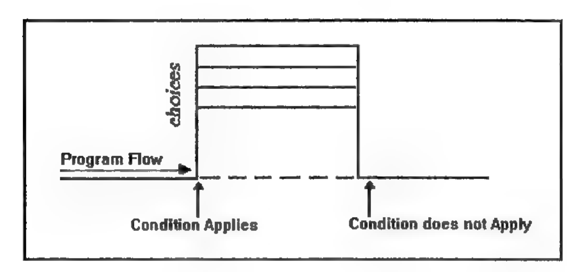
في بعض الأحيان يراد تنفيذ بعض العمليات التي ترتبط بشرط معين (Condition) . وذلك عندما يراد مثلا ادخال زيادة في مرتب الموظفين الذين تزيد مدة خدمتهم عن ١٠ سنوات . ولتنفيذ ذلك يتيح البرنامج طريقتين لتنفيذ هذه العملية ، الأولى باستخدام (IF-ENDIF) وذلك عندما يكون هناك احتمالان فقط للشرط . انظر الشكل (١٢ - ٤)



شكل (١٢ - ٤)

أمـــا اذا كـــانت هـــــناك عـــدة احتمـــالات ، فيستخـــدم الأمــــر (DO CASE-ENDCASE) . وذلك عندما يراد مثلا عرض قائمة اختيارات على الشاشة والتفرع الى البرنامج الذى ينفذ الاختيار المطلوب للمستخدم .

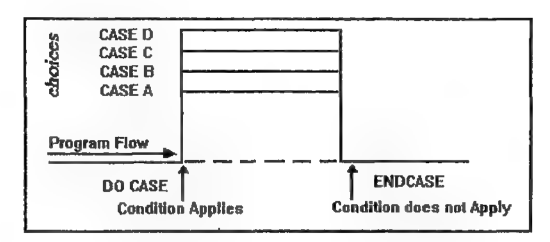
انظر الشكل (١٢ - ٥)



شكل (١٢ - ٥)

والقوائم تستخدم في معظم برامج ادارة قواعد البيانات وتبدأ عادة بقـــائمة رئيسيـــة (Main Menu) تتفرع الى قوائم أخـرى فـرعية (Submenus) تظهر عند اختيار المســـتخدم لأحـــــد اختيـــارات القائمـــة الرئيسيـــة . ويستخـــدم الأمـــر (DO CASE-ENDCASE) في التفرع من البرنامج بناء على الشرط الذى يلى الأمر (DO CASE) . حيث يتم تنفيذ كل مجموعة من الأوامر حسب القيمة الموجودة بعد كلمة (CASE) .

أنظر الشكل (١٢ - ٦)



شكل (١٢ - ٦)

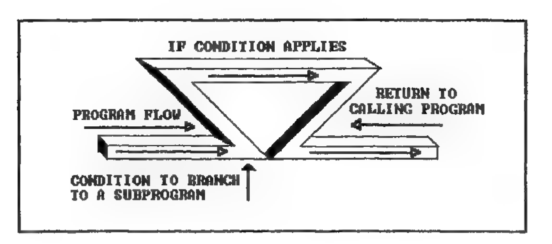
١٢ - ٦ - ٢ التفرع الى برنامج فرعى

كما سبق الايضاح في المثال الخاص بوصفة الطهي (Cooking Recipe)، تكون هناك مجموعة من الارشادات والتعليمات لطهي نوع معين من الأطعمة وهذه الارشادات تكون في الواقع برنامجا مثل برامج الحاسب . وعندما يراد مثلا عمل فلفل مقلي ضمن الوجبة فان الطاهي يلجأ الى وصفة أخرى خاصة بقلي الفلفل وينفذها ثم يعود الى الوصفة الأولى ويكمل تنفيذ التعليمات الموجودة بها . وهذه الوصفة الأخرى الخاصة بقلي الفلفل تمثل برنامجا فرعيا متفرعا من البرنامج الأول .

وعند كتابة البرنامج يمكن التفرع الى برامج أخرى لتنفيذ العمليات التي يحتاجها المستخدم . فمثلا عندما تكون هناك قائمة اختيارات يختار منها المستخدم اختيارا معينا ، فان البرنامج الرئيسي ينتقل الى برنامج آخر ينفذ هذا الاختيار .

وعند الانتهاء من تنفيذ البرنامج الفرعي يعود البرنامج مرة أخرى الى نفس المكان الذى انتقل منه . وهذه الخاصية تعطى قوة ومرونة كبيرة للبرنامج لأنها تسمح بتقسيم أي برنامج كبير الى عدة برامج صغيرة (Modules)مما يسهل اختبار كل برنامج صغير وتصحيحه مستقلا عن باقي البرامج .

انظر الشكل (١٢ - ٧)



شكل (١٢ - ٧)التفرع الى برنامج فرعى

ولتنفيذ هذه العملية يستخدم الأمر (DO) وبعده اسم البرنامج المطلوب تنفيذه . ويكتب هذا الأمر داخل البرنامج الرئيسي الذى يسمى فى هذه الحالة برنامج الاستدعاء (Calling Program). وعندما يجد البرنامج الأمر (DO) فانه يذهب الى البرنامج الفرعى (Module) وينفذه ثم يعود الى الأمر التالى للأمر (DO) مباشرة ويكمل تنفيذ البرنامج . وسوف يتم شرح هذا الأمر بالتفصيل فيما بعد.

١٢ - ٦ - ٣ الحلقة التكرارية (LOOP)

فى بعض الأحيان يراد تنفيذ مجموعة من الأوامر عددا من المرات يتوقف على تحقق شرط معين . وهذه المجموعة من الأوامر تسمى الحلقة التكرارية (Loop). وهى تبدأ بالأمر (DO WHILE) وتنتهى بالأمر (ENDDO). وهذان الأمران يمثلان حدود الحلقة التكرارية .

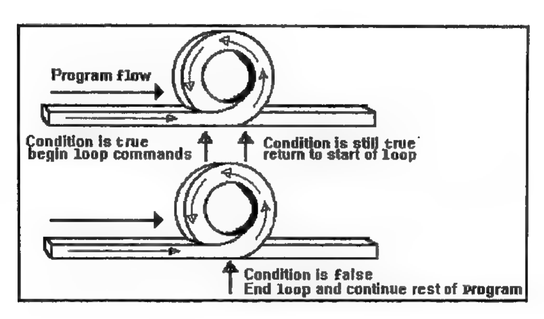
فمثلا عندما يراد عرض بيانات المتزوجين فقط فى قاعدة البيانات الخاصة بالموظفين يستخدم الأمر التالى :

```
DO WHILE Married
```

حيث (Married) هو اسم الحقل المنطقى الذى يحدد اذا كان الموظف متزوجا أو غير متزوج .

وعند كتابة الأمر بهذا الشكل فان البرنامج ينفذ الأوامر التالية له طالما كان الموظف متزوجا . لأن الحقل في هذه الحالة يكون صحيحا (True) . أما الموظف غير المتزوج فان الحقل الخاص به يكون غيرصحيح (False) وبالتالى لا يتم عرض بيانات الموظفين الغير متزوجين .

انظر الشكل (١٢ - ٨)



شكل (١٢ - ٨)

ويلاحظ من هذا الشكل أن تدفق البرنامج يصل الى شرط معين مطلوب تحقيقه . فاذا تحقق الشرط يتم تنفيذ أوامر الحلقة . وعندما يصبح الشرط غير صحيح يتوقف تنفيذ الحلقة ويستمر تدفق البرنامج فى اتجاهه المعتاد .

١٢ - ٧ الاعداد للبرنامج

قبل البدء في كتابة البرنامج يجب أولا دراسة مخرجات البرنامج المطلوبة مثل التقارير المطبوعة أو المعروضة على الشاشة . وهذه المرحلة تتطلب اشتراك المستخدم مع مصمم البرنامج لتحديد المخرجات المطلوبة وكذلك لتحديد الآتى :

١ - الهدف من البرنامج .

٢ - كيفية تحقيق البرنامج لهذا الهدف .
٣ - نوع وشكل شاشات المساعدة (Help Screens) .
٤ - تحديد امكانية تحسين البرنامج وتطويره .
٥ - تحديد مصـــدر مدخلات البرنامج واذا كانت عن طريق لوحة المفاتيح أو من ملفات معينة .

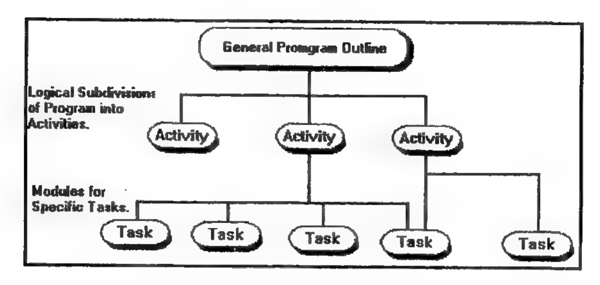
١٢ - ٨ التصميم من أعلى الى أسفل (Top-Down Design)

يبدأ تصميم البرنامج بتحديد الخطوط العامة (General Outline) التى توضح خطوات تصميم البرنامج بالتسلسل المنطقى الذى يؤدى الى تحقيق الهدف النهائى للبرنامج . فمثلا عندما يراد كتابة برنامج يقوم بعرض بيانات سجل معين فى قاعدة بيانات معينة ، يتـم كتابة الخـطوات التالية لتحديد الخطوط العامة للبرنامج :

١ - تجهيز محيط التشغيـــل (Working Environment) وذلك بتحديـد الفهرس الفرعى المستخدم وفتح الملفات المستخدمة ومسح الشاشة و... الخ .
٢ - تحديد السجل المطلوب عرضه .
٣ - تحديد الحقول المطلوب عرضها فى السجل .
٤ - عرض الحقول المطلوبة فى السجل .
٥ - سؤال المستخدم اذا كان يريد عرض سجل آخر أم لا .
٦ - اذا كان المستخدم يريد عرض سجل آخر يتم اعادة الخطوات ٢، ٣، ٤.
٧ - اذا كان المستخدم لايريد عرض سجل آخر يتم اغلاق جميع الملفات والعودة الى مشيرة النقطة (Dot Prompt) .

وبعد تحديد الخطوط العامة (General Outline) يتم تحديد الوظائف الرئيسية للبرنامج والتى تحقق هذه الخطوات . ثم يتم فصل هذه الوظائف بالتدريج الى وظائف أصغر فأصغر ثم كتابة البرامج التى تؤدى الى تنفيذ هذه الوظائف الصغيرة . هـذه البرامج الصغيرة التى تسمى (Modules) يؤدى كل منها وظيفة محددة ومستقلة من وظائف البرنامج الرئيسى . كما أن هذا البرنامج الصغير (Module) يمكن أن يتكـرر استخدامه داخل البرنامج الرئيسى عندما يتطلب البرنامج تنفيذ الوظيفة الخاصة عدة مرات . كما يمكن استخدام نفس هذا البرنامج الفرعى مع برامج رئيسية أخرى . وتسمى هذه الطريقــة فى كتابـة البرامـج البرمـجـة التركيبية (Structural Programming) .

وهذه الطريقة تمتاز عن الطريقة التقليدية بسهولة كتابة كل برنامج واختباره منفصلا . كما أن هذه البرامج الصغيرة يمكن استخدامها فى برامج أخرى متعددة لتحقيق وظائف محددة فى هذه البرامج . كما يمكن تجميع مجموعة كبيرة من هذه البرامج الصغيرة فى مكتبـــة برامـــج (library) يمكن استخدامها عند الحاجة . انظر الشكل (١٢ - ٩)



شكل (١٢ - ٩)

ومن فوائد هذه الطريقة التركيبية أيضا أنها تساعد على تطوير البرنامج فيما بعد أو توسيعه (Expansion) بسهولة . حيث يكفى فى هذه الحالة اضافة برامج فرعية جديدة تحقق وظائف اضافية للبرنامج الرئيسى .

١٢ - ٩ كتابة الملاحظات فى البرنامج (Comments)

من المهم جدا وضع ملاحظات داخل البرنامج لتوضيح خطوات البرنامج ، وذلك حتي يسهل بعد ذلك تصحيحه أو تطويره . وسطر الملاحظات يبدأ بكلمة (NOTE) أو الحرف (*) . وعندما يقوم البرنامج المترجم (Interpreter) بترجمة شفرة البرنامج فانه يمر على سطور الملاحظات دون ترجمتها ، ولكن هذه الملاحظات تظهر عند عرض سطور البرنامج على الشاشة عند تصحيحه . ومن الملاحظات السطور التالية مثلا :

```
*       This is a comment line
Note this line is not a command line
```

ويمكن اضافة الملاحظات في نفس سطر الأمر لتوضيح وظيفة هذا الأمر وذلك بكتابة الحرفين (&&) قبل الملاحظات كالآتى :

```
DO Menu && Run the Menu Program
```

ملاحظة

يجب فصل الحرفين (&&) عن الأمر الموجود في السطر بمسافة واحدة (Space) على الأقل .

وهذه الملاحظات تسهل تتبــع أوامر البرنامج وتسلسله المنطقي واكتشاف أى أخطاء . كما أنها تساعد أي شخص غير الشخص الذى قام بكتابة البرنامج على تعديل البرنامج أو تطويره أو اكتشاف أى أخطاء أو مشاكل به .

الباب الثالث عشر

تركيب البرنامج

Program Structure

يمكن تقسيم البرنامج المكتوب الى أربعة أقسام رئيسية وهى المقدمة وأوامر التجهيز (Set Up) وأوامر البرنامج وأوامر الخروج .

١٣ - ١ المقدمة

وهى اختيارية يمكن كتابتها أو عدم كتابتها ، ويتم فيها كتابة اسم مصمم البرنامج وتاريخ تصميمه وأى معلومات أخرى مطلوب اضافتها . وتستخدم لكتابة سطورها كلمة (NOTE) أو الحرف (*) فى بداية كل سطر .

١٣ - ٢ أوامر التجهيز (Setup)

وهى الأوامر التى تؤدي الى تجهيز محيط التشغيل (Working Invironment) لاستقبال أوامر البرنامج . وبعض هذه الأوامر يبدأ بالأمر (SET) مثل الأوامر التالية :

```
SET TALK OFF
SET DEFAULT TO
```

وسوف يتم دراستها بالتفصيل فيما بعد .

وهناك أوامر أخرى تستخدم فى فتح ملفات قاعدة البيانات وملفات الفهرس (Index) والبحث (Query) و .. الخ . وهناك أوامر أخرى تستخدم فى انشاء متغيرات الذاكرة (Memory Variables) التي تستخدم فى التخزين المؤقت للمدخلات والمخرجات . وسوف يتم دراستها فيما بعد .

وأوامر التجهيز عادة يكتب معظمها فى البرنامج الرئيسي ، وتصبح فى هذه الحالة مؤثرة فى البرنامج الرئيسى والبرامج الفرعية فى نفس الوقت . كما أن هناك بعض أوامر التجهيز التي تكتب فى برامج فرعية معينة ، وفى هذه الحالة لاتؤثر الا فى هذه البرامج الفرعية فقط .

١٣ - ٣ أوامر البرنامج

وهى الأوامر الرئيسية فى البرنامج التى تؤدى المهام الرئيسية مثل استقبال المدخلات (Input) من المستخدم ، وعرض البيانات وتعديل البيانات وانشاء المخرجات (Output) . وهي تشمل أوامر التحكم والتكرار (Looping) والتفرع الى برامج فرعية أخرى لتنفيذ المهام المطلوبة من البرنامج .

١٢ - ٤ أوامر الخروج

وهي الأوامر التي تؤدى الى الخروج من البرنامج والعودة الى مشيرة النقطة (Dot Prompt) . وهي تشمل أوامر اغلاق جميع الملفات المفتوحة (Closing Commands) للمحافظة على وحدة وتكامل قاعدة البيانات (Database Integrity) ، ومنها الأوامر التالية :

```
CLOSE DATABASES
CLEAR ALL
USE
```

ثم تأتى أوامر الخروج مثل :

```
RETURN
QUIT
```

والأمر (RETURN) اذا كتب في البرنامج الرئيسى فانه يؤدى الى العودة الى مشيرة النقطة (Dot Prompt) ، أما اذا كتب فى برنامج فرعي فانه يؤدى الى العودة الى البرنامج الذى قام باستدعائه (Calling Program) . أما الأمر (Quit) فانه يؤدى الى الخروج من برنامج (DBase III +) والعودة الى نظام التشغيل .

١٢ - ٥ استخدام الأمر (DO)

يستخدم الأمر (DO) كما سبق الايضاح في التفرع من البرنامج الرئيسى الى برنامج فرعي . كما يمكن أيضا التفرع من برنامج فرعى الى برنامج فرعي آخر . وهذا التفرع يكون غير مشروط (Uncoditional) . فعندما يقابل مترجم البرنامج (Interpreter) الأمر (DO) فانه ينتقل الى البرنامج الفرعي المحدد بالاسم الموجود بعد الأمر (DO) . ويستمر فى تنفيذ أوامر البرنامج الفرعى حتى يصل الى الأمر (RETURN) الذى يؤدى الى الرجوع الى البرنامج القائم بالاستدعاء (Calling Program) .

فمثلا يمكن أن يحتوى برنامج على الأوامر التالية :

```
CLEAR            && clears the screen
USE CADETS       && opens cadets.dbf file
DO C_EDIT        && Branches to C_EDIT.prg
DO C_DEL         && Branches to C_DEL.prg
```

حيث (C_EDIT) هو برنامج فرعى يـؤدى الى تصحيح البيانات في ملف قاعدة

البيانات و (C_DEL) هو برنامج فرعي آخر يؤدى الى مسح السجلات المطلوب الغاؤها من ملف قاعدة البيانات . ويلاحظ هنا كتابة اسم البرنامج الفرعي بدون الامتداد (Extension) الذى يضيفه البرنامج آليا ويكون دائما (.prg) .

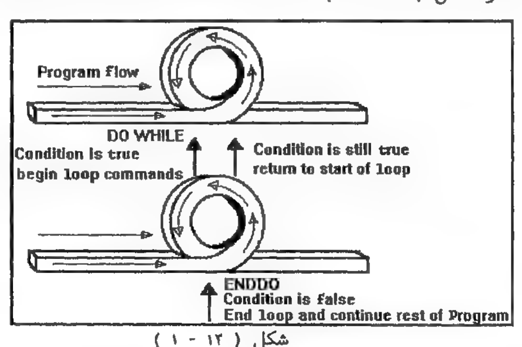
وعندما يصل البرنامج الى الأمر (DO C_EDIT) مثلا ، فانه ينتقل الى برنامج (C_EDIT) وينفذ أوامره ثم يعود الى البرنامج مرة ثانية عندما يقابل الأمر (RETURN) الذى يكون آخر أمر فى البرنامج الفرعى .

ملاحظة

يمكن استخدام الأمر (RETURN TO MASTER) للرجوع مباشرة الى البرنامج الرئيسى بصرف النظر عن عدد البرامج الفرعية التى تم التفرع اليها . وهذا يؤدى الى سرعة الرجوع الى البرنامج الرئيسى دون المرور على عدة برامج فرعية .

١٢ - ٦ استخدام الأمر (DO WHILE)

يستخدم هذا الأمر كما سبق الايضاح فى تكرار تنفيذ مجموعة من الأوامر عددا من المرات يتوقف على تحقق شرط معين يكتب بعد الأمر حيث يتم اختبار الشرط فى بداية الحلقة التكرارية (Loop) ، فاذا كان صحيحا (TRUE) يتم تنفيذ جميع الأوامر التاليـة للأمر (DO WHILE) ثم يتوقف عند الأمر (ENDDO) الذى يؤدى الى العودة الى بداية الحلقة التكرارية . ثم يتم اختبار الشرط مرة ثانية ، فاذا كان صحيحا ، يتم تنفيذ أوامر الحلقة وهكذا . وعندما يصبح الشرط غير صحيح (FALSE) يتوقف تنفيذ الحلقة وينتقل البرنامج الى الأمر التالى للأمر (ENDDO) . أنظر الشكل (١٢ - ١)



شكل (١٢ - ١)

فمثلا يمكن أن يحتوى برنامج على الأوامر التالية :

```
DO WHILE Age <= 40
     --------
     --------
     --------          Commands
     --------
     --------
ENDDO
```

فى هذه الحالة يتم تنفيذ أوامر الحلقة طالما كان العمر أقل من أو يساوى ٤٠ سنة . ويستمر تنفيذ أوامر الحلقة التكرارية حتى يصبح العمر أكبر من ٤٠ سنة . وفى هذه الحالة ينتقل البرنامج الى الأوامر التالية للأمر (ENDDO) .

كما يمكن أن يحتوى برنامج آخر على الحلقة التكرارية التالية :

```
DO WHILE .NOT. EOF()
     DISPLAY Name, Address, Age
     SKIP
ENDDO
```

يلاحظ فى هذه الحالة استخدام العامل المنطقى (NOT) ، وكذلك استخدام الداله EOF() ، وهى تعنى (End of file) أى نهاية ملف قاعدة البيانات المفتوح .

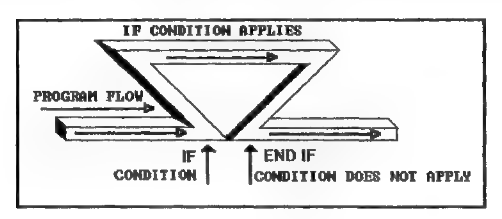
وتؤدى الحلقة التكرارية (Loop) الى عرض حقول الاسم والعنوان والعمر لجميع السجلات حتى يصل المؤشر الى آخر سجل فى الملف .

وتبدأ الحلقة باختبار وصول المؤشر الى نهاية الملف ، فاذا لم يصل الى نهاية الملف يصبح الشرط صحيحا وبالتالى يتم تنفيذ أوامر الحلقة ، ويتم عرض حقول الاسم والعنوان والعمر لهذا السجل . ثم يؤدى الأمر (SKIP) الى الانتقال الى السجل التالى . كما يؤدى الأمر (ENDDO) الى العودة الى أول أمر في الحلقة التكرارية ويتم اختبار الشرط مرة ثانية . وهكذا يتكرر تنفيذ الحلقة التكرارية حتى يصل البرنامج الى نهاية الملف .

ويلاحظ فى هذه الحلقة ادخال أوامر الحلقة قليلا الى الداخل . ويسمى ذلك (Indentation) ويستخدم عادة في البرنامج لتوضيح الأوامر الخاصة بالحلقة التكرارية خاصة عندما تتعدد الحلقات التكرارية وتصبح متداخلة (Nested) .

١٢ - ٧ استخدام الأمر (IF-ENDIF)

يستخدم هذا الأمر عندما يراد تنفيذ أمر أو مجموعة من الأوامر مرة واحدة عند تحقق شرط معين . حيث يتم اختبار الشرط التالى للأمر (IF) فاذا تحقق ، يتم تنفيذ الأوامر التالية له . أما اذا لم يتحقق ، فيتم الانتقال مرة واحدة الى الأمر (ENDIF) وتنفيذ الأوامر التالية له أى أنه فى هذه الحالة يتخطى الأوامر المحصورة بين الأمرين (IF,ENDIF) . أنظر الشكل (١٢ - ٢).



شكل (١٢ - ٢)

فمثلا يمكن أن يحتوى برنامج على الأوامر التالية :

```
IF MARK < 50
     ? "Failed"   & & Display the message
ENDIF
```

وفى هذا المثال يتم اختبار الشرط الموجود بعد الأمر (IF) . فاذا كانت الدرجة (MARK) أصغر من ٥٠ درجة تظهر رسالة (Failed) . واذا كانت الدرجة أكبر من أو تساوى ٥٠ درجة لاتظهر هذه الرسالة . والحرف (?) هو أمر من أوامر (+ DBase III) يستخدم فى عرض رسائل على الشاشة .

ويمكن زيادة امكانيات هذا الأمر عن طريق استخدام (ELSE) . فيمكن مثلا كتابة الأوامر التالية فى برنامج :

```
IF MARK < 50
     ? "Failed"
ELSE
     ? "Passed"
ENDIF
```

ويؤدى ذلك الى ظهور رسالة (Failed) فى حالة تحقق الشرط وظهور رسالة (Passed) فى حالة عدم تحقق الشرط .

كما يمكن استخدام مجموعة متداخلة (Nested) من أوامر (IF) كما سيتم الايضاح فيما بعد فى الجزء الخاص بالتداخل (Nesting) .

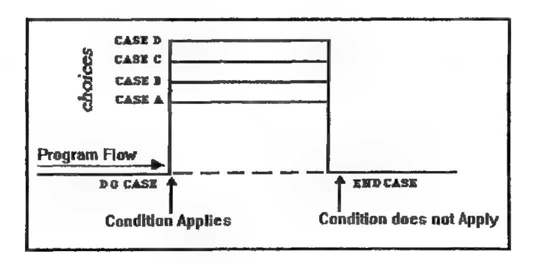
ويمكن استخدام المعاملات المنطقية فى الجمع بين عدة شروط كما يلاحظ من مجموعة الأوامر التالية :

```
IF Age > 40 .AND. .NOT. Married
     DO Payment
ENDIF
```

حيث يتم تنفيذ البرنامج (Payment) فقط للموظفين الذين يزيد عمرهم عن ٤٠ سنة وغير متزوجين . ويجب ملاحظة أن المعامل المنطقى يكون محصورا بين نقطتين . كما يجب فصل كل معامل عن المعامل الآخر بمسافة خالية (Space) على الأقل .

١٣ - ٨ استخدام الأمر (DO CASE - ENDCASE)

عندما يراد التفرع الى برامج مختلفة بناء على شروط معينة يستخدم الأمر (DO CASE) . فعندما يقابل البرنامج الامر (DO CASE) فانه يذهب الى كل سطر يبدا بكلمة (CASE) و يختبر الشرط الموجود معها ، فاذا تحقق الشرط ينفذ البرنامج أو الأوامر التالية لهذا الشرط ، واذا لم يتحقق الشرط فانه يذهب الى باقى السطور التى تبدأ بكلمة (CASE) ، ويختبر الشروط الخاصة بها . ويستخدم هذا الأمر بصفة خاصة فى القوائم ، حيث يتم التفرع من البرنامج الرئيسى الى برامج فرعية بناء على اختيار المستخدم . انظر الشكل (١٣ - ٢)



شكل (١٣ - ٢)

وتنتهي هذه المجموعة بالأمر (ENDCASE) الذي يؤدى الى الانتقال الى الأوامر التى تلى هذا الأمر .

كما يمكن استخدام الأمر (OTHERWISE) مع هذا الأمر لتنفيذ أمر معين فى حالة عدم تحقق أى شرط من الشروط التى تلى كل أمر (CASE) .

ولتوضح ذلك يمكن ملاحظة الأوامر التالية :

```
DO CASE
     CASE Choice = "A"
          DO MEdits
CASE Choice = "B"
          DO MDel
     CASE Choice = "C"
          DO MRep
     OTHERWISE
          DO Leave
ENDCASE
```

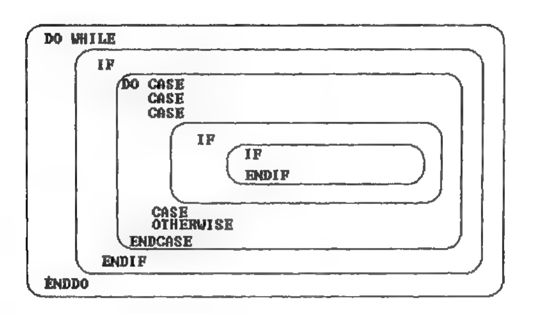
وفى هذا المثال يتم تنفيذ برنامج (MEdit) عند اختيار المستخدم للحرف (A). ويتم تنفيذ برنامج (MDel) عند اختيار المستخدم للحرف (B) . ويتم تنفيذ برنامج (MRep) عند اختيار المستخدم للحرف (C) . ويتم تنفيذ برنامج (Leave) عند اختيار المستخدم لأى حرف آخر غير الحروف (A,B,C) .

١٢ - ٩ التداخل (Nesting)

يمكن استخدام أوامر التحكم السابق شرحها بالتداخل فيما بينها وذلك كالآتى مثلا :

```
IF MARRIED
     DO CASE
          CASE Age > 20 .AND. Age < 30
          ------
          ------        Commands
          ------
          CASE Age > 30 .AND. Age < 40
          ------
          ------        Commands
          ------
     ENDCASE
ENDIF
```

ويلاحظ في هذا المثال أن الحلقة الخاصة بالأمر (DO CASE) داخله بالكامل فى الحلقة الخاصة بالأمر (IF) . ويلاحظ أيضا فى هذه الحالة أن الأمر (ENDCASE) يجب أن يسبق الأمر (ENDIF) والا يصبح التداخل غير صحيح . كمايفيد ادخال الكتابة الى الداخل (Indentation) فى توضيح حدود كل حلقة . أنظر الشكل (١٢ - ٤) .



شكل (١٢ - ٤)

ملاحظة

يجب دائما التأكد من أن كل أمر من الأوامر (DO WHILE, IF, DO CASE) له أمر انهاء خاص به (ENDDO, ENDIF, ENDCASE) على الترتيب . كما يجب التأكد من عدم اختلاف ترتيب أوامر الانهاء عن ترتيب أوامر البداية . ويجب ملاحظة أن أوامر الانهاء تكتب كلمة واحدة وليست كلمتين أى (ENDDO, ENDIF, ENDCASE) بدون مسافات بينها .

ويمكن استخدام الملاحظات فى تحديد أى أمر نهاية يتبع أى أمر بداية . ويمكن توضيح ذلك من البرنامج التالى :

```
DO WHILE .NOT. EOF()
     LIST Name , Age
     SKIP
ENDDO  && while not EOF()
```

هذه الملحوظة التى تلى الحرفين (&&) تؤدى الى توضيح أن أمر الانهاء (ENDDO) يختص بأمر البداية المحدد فى الملحوظة .

١٢ - ١٠ استخدام الأمر (LOOP)

يستخدم الأمر (LOOP) للرجوع من أى مكان داخل الحلقة التكرارية الى أول الحلقة دون استكمال أوامر الحلقة . ويمكن توضيح ذلك من خلال الأوامر التالية :

```
DO WHILE .NOT. EOF()
     IF Mark < 50
          ? "FAILED"
          SKIP
          LOOP
     ENDIF &&  Mark < 50
          ----------
          ----------
          ----------        Commands
          ----------
          ----------
ENDDO && WHILE .NOT. EOF()
```

وفى هذا المثال يتم اختبار درجة الطالب ، فاذا كانت أقل من ٥٠ درجة تظهر الرسالة (FAILED) ، ويتم الانتقال الى السجل التالى . ويؤدى الأمر (LOOP) الى الرجوع الى الحلقة لاختبار درجة الطالب الجديد . ويستمر ذلك حتى يتم الوصول الى أى طالب درجته أكبر من ٥٠ درجة حيث يتخطى البرنامج مجموعة الأوامر المحصورة بين (IF, ENDIF) وينفذ مجموعة الأوامر التالية . وعندما يصل البرنامج الى الأمر (ENDDO) يتم الانتقال الى أول الحلقة التكرارية مرة أخرى وتتكرر هذه العملية .

١٢ - ١١ الخروج من الحلقة التكرارية

يستخدم الأمر (EXIT) فى الخروج من الحلقة التكرارية والانتقال الى الأوامر التى تلى الأمر (ENDDO) . وهو يختلف عن الأمر (LOOP) حيث أن الأمر (LOOP) يؤدى الى الرجوع الى أول الحلقة التكرارية والبدء فى تنفيذها من جديد . أما الأمر (EXIT) فانه لايؤدى الى الرجوع الى أول الحلقة التكرارية ولكنه يؤدى الى الخروج منها وتنفيذ الأوامر التى تلى الأمر (ENDDO) .

مــلاحظة

ما سبق ذكره في هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل (DBase IV) ، (FoxBase +) ، (FoxPro) .

الباب الرابع عشر

متغيـــرات الذاكرة

Memory Variables

متغيرات الذاكرة هي متغيرات من نوع خاص يتم فيها تخزين البيانات تخزينا مؤقتا خارج ملف قاعدة البيانات . وعن طريق هذه المخازن المؤقتة يمكن اجراء العمليات الحسابية أو أى عمليات على بيانات الملف دون تغيير بيانات الملف مباشرة .

و يجب الايزيد طول اسم المتغير عن عشرة حروف . ويمكن أن تحتوى على حروف أو أعداد أو شرطة سفلية (Underscore) . ولكن يجب أن تبدأ بحرف ولاتحتوى على مسافات . وهناك أربعة أنواع من متغيرات الذاكرة ، وهى متغيرات حرفية (Character) ، ومتغيرات تاريخية (Date) ومتغيرات عددية (Numeric) ، ومتغيرات منطقية (Logical) ، ولكن ليست هناك متغيرات لتخزين الملاحظات (Memo) .

ويمكن أن يصل عدد المتغيرات المستخدمة فى البرنامج الى ٢٥٦ متغيرا بحيث لايزيد حجمها الكلى عن ٦٠٠٠ حرفا . ويمكن زيادة حجمها بكتابة الأمر (MVARSIZ) في ملف المواصفات (Config.sys) يليه عدد الحروف المراد استخدامها . وذلك حسب الذاكرة المؤقتة المتاحة فى الجهاز .

ملاحظة

عندما تكون هناك حقول في ملف قاعدة البيانات لها نفس الاسم مثل متغيرات الذاكرة فان الأسبقية تكون لحقول الملف عندما يشار الى هذا الاسم . فمثلا عندما يكون هناك حقل مثل (Name) فى ملف قاعدة البيانات ، وتم انشاء متغير ذاكرة بنفس الاسم . فان أى بيانات يتم ادخالها الى المتغير (Name) تذهب الى الحقل وليس الى متغير الذاكرة . وللتغلب على ذلك يتم اضافة الحرف (M) قبل اسم المتغير حتى يصبح مختلفا عن اسم الحقل وفى نفس الوقت يعطى معنى محتويات الحقل ، أى يصبح (MName) .

ويراعى أيضا في اختيار أسماء متغيرات الذاكرة أن تكون مختلفة عن الأسماء المحجوزة لبرنامج (+DBase III) مثل (Continue) مثلا .

ارجع الى الكتاب الثانى للتعرف على جميع الأوامر والدوال المستخدمة فى برنامج (DBase III+).

١٤ - ١ أنواع متغيرات الذاكرة (Memory Variables)

كما سبق الايضاح هناك أربعة أنواع من متغيرات الذاكرة ، و يتم شرحها فى الاجزاء التالية .

١٤ - ١ - ١ المتغيرات الحرفية (Character)

وتستخدم لتخزين المدخلات الحرفية . ويمكن ادخال بيانات حرفية تحتوي على كود الآسكي الخاص بالحروف باستخدام الدالة (CHR) كما سيتم الايضاح فيما بعد .

والمتغيرات الحرفية يمكن أن تحتوي على عدد حروف بحد أقصى ٢٥٤ حرفا . وتحتل في الذاكرة مساحة تخزينية تساوي عدد الحروف المخزنة مضافا اليه حرفين .

١٤ - ١ - ٢ المتغيرات التاريخية (Date)

وهي تستخدم لتخزن التاريخ . وحجم المتغير التاريخي ٨ حروف دائما . ويتم ادخال التاريخ بالطريقة الأمريكية (mm/dd/yy) أي الشهر ثم اليوم ثم السنة كما يمكن استخدام صورة أخرى باستخدام الأمر (SET DATE) كما سيتم الايضاح فيما بعد .

ويمكن اجراء عمليات حسابية على التاريخ مثل طرح تاريخ من تاريخ للحصول على عدد الأيام المحصورة بينهما . كما يمكن طرح عدد الأيام من تاريخ معين للحصول على تاريخ آخر .

١٤ - ١ - ٣ المتغيرات العددية (Numeric)

وتستخدم لتخزين الأعداد التي يمكن اجراء عمليات حسابية عليها . ويمكن أن تحتوي على ١٥ رقما بما فيها الأرقام العشرية (Decimal) التي يجب الا يزيد عددها عن ٩ أرقام عشرية .

١٤ - ١ - ٤ المتغيرات المنطقية (Logical)

وهي متغيرات يتم فيها تخزين حرف واحد فقط يمثل حالة البيان اذا كان صحيحا (True) ، أو غير صحيح (False) . هذا الحرف يكون أحد الحروف التالية (T, F, Y, N) .

١٤ - ٢ انشاء متغيرات الذاكرة

يتم انشاء متغيرات الذاكرة بمجرد تخزين بيانات فيها . ويستخدم لذلك الأمر (Store) . كما أن البرنامج يقوم بتحديد نوع المتغير حسب نوع البيانات التي يتم ادخالها .

ويمكن انشاء متغيرات الذاكرة بطريقة أخرى وذلك بكتابة اسم المتغير أولا وبعده علامة التساوى ثم البيانات المطلوب تخزينها فيه . وفى الأجزاء التالية يتم عرض عدة أمثلة توضح طريقة انشاء متغيرات الذاكرة بالطريقتين .

١٤-٢ -١ انشاء المتغيرات المنطقية (Logical Variables)

الأمر التالى يؤدى الى انشاء متغير يسمى (Married) ويتم تخزين القيمة (.T.) أى (True) فيه :

```
STORE .T. To Married
```

كما يمكن استخدام الأمر التالى ليحقق نفس النتيجة :

```
Married = .T.
```

وفى هذه الحالة تصبح البيانات يمين علامة التساوى هى محتويات المتغير ويمكن تغيير هذه المحتويات بعد ذلك حسب الحاجة .

١٤-٢-٢ انشاء المتغيرات الحرفية (Character Variables)

الأمر التالى يؤدى الى انشاء متغير ذاكرة اسمه (mname) ويتم تخزين الاسم (Mohamed) فيه :

```
STORE "Mohamed" TO mname
```

ويلاحظ في هذه الحالة وضع الحروف (String) بين علامات تنصيص (Quotation) . أما اذا كانت الحروف المطلوب ادخالها فى المتغير تحتوى على علامات تنصيص داخلها ، يتم استخدام علامات تنصيص مختلفة عن العلامات داخل الحروف (String) . فمثلا يمكن استخدام الأمر التالى :

```
STORE "That's incorrect..try again" TO mmessage
```

ويمكن انشاء متغير ذاكرة لايحتوى على أى بيانات باستخدام الدالة(() Space) كالآتى :

```
STORE SPACE(20) TO mname
```

وهذا يؤدى الى حجز متغير اســمه (mname) طوله ٢٠ حرفا وليس فيه أى بيانات . وذلك حتى يمكن استخدامه بعد ذلك فى تخزين بيانات الاسم .

١٤ - ٢ - ٣ انشاء المتغيرات التاريخية (Date Variables)

لانشاء متغير تاريخى يمكن استخدام دالة التحويل من الحروف الى التاريخ (Character to date conversion function) التى تختصر الى (CTOD()). هذه الدالة تؤدى الى تحويل الحروف التى يكتبها المستخدم ممثلة تاريخا معينا الى قيمة مقابلة يخزنها البرنامج . فمثلا لانشاء متغير تاريخى اسمه (Birthday) يتم كتابة الأمر التالى :

```
birthday = CTOD ('20/1/49')
```

ويمكن بعد ذلك تغيير محتويات هذا المتغير بأى تاريخ آخر . كما يمكن انشاء متغير تاريخى ليس به أى تاريخ بكتابة الأمر التالى :

```
birthday = CTOD('/ /')
```

كما يمكن استخدام الأمر (STORE) فى انشاء المتغير التاريخى كالآتى :

```
STORE CTOD('/ /') To birthday
```

كما يمكن استخدام دالة التاريخ DATE() فى ادخال تاريخ اليوم الحالى كالآتى :

```
STORE DATE() TO today
```

١٤-٢- ٤ انشاء المتغيرات العددية (Numeric Variables)

لانشاء متغيرات عددية يكفى ادخال صفر فى التغير كالآتى مثلا :

```
STORE 0 TO number
```

مع ملاحظة أن هذا المتغير يقبل رقما صحيحا فقط . أما اذا أريد انشاء متغير يقبل كسرا عشريا ، فيتم وضع نقطة الكسر العشرى (Decimal Point) ، وذلك كالآتى :

```
STORE 0.00 TO number
```

في هذه الحالة يتم انشاء متغير عددي يقبل رقمين عشريين . ويمكن عن طريق زيادة عدد الأصفار بعد العلامة العشرية ، زيادة عدد الأرقام العشرية التى يقبلها المتغير العددى .

كما يمكن انشاء عدة متغيرات عددية من نفس النوع باستخدام أمر واحد. وذلك كالآتى مثلا :

```
STORE 0.00 TO num1 , num2 , num3
```

ملاحظة

عند اختيار اسم متغير الذاكرة يراعى أن يكون معبرا عن محتويات هذا المتغير . كما يراعى أيضا عندما يكون اسم المتغير مطابقا لاسم حقل فى ملف قاعدة البيانات أن يضاف الحرف (M) قبل اسم المتغير حتى يتم تمييزه عن اسم الحقل . فمثلا اذا كان هناك حقل اسمه (Name) فى قاعدة البيانات ، يمكن انشاء متغير ذاكرة اسمه (MName) أو (M_Name) .

١٤ - ٣ أهمية متغيرات الذاكرة

تستخدم متغيرات الذاكرة كما سبق الايضاح كمخزن مؤقت للبيانات التى يتم ادخالها الى ملف قاعدة البيانات أو اخراجها الى أجهزة الاخراج المختلفة . وهذه العملية فى منتهى الأهمية فى كتابة البرامج ، وذلك لأنها تعطى الفرصة للمستخدم لاختبار البيانات التى يقوم بادخالها قبل تخزينها فى قاعدة البيانات . فمثلا عندما يراد ادخال بيانات فى حقل الاسم (Name) فى قاعدة بيانات معينة يتم اتباع الخطوات التالية :

١ - يتم انشاء متغير ذاكرة حرفى نسميه مثلا (MName) بحيث يكون بنفس طول حقل الاسم (Name) الخاص بملف قاعدة البيانات .
٢ - يتم استخدام هذا المتغير فى تخزين الاسم الذى يقوم المستخدم بادخاله تخزينا مؤقتا .
٣ - يتم عرض الاسم الذي أدخله المستخدم حتى يتأكد أنه الاسم المطلوب ، ويتم سؤال المستخدم اذا كان الاسم صحيحا أم لا .
٤ - اذا كان رد المستخدم (No) يتم اعطاؤه الفرصة لتصحيح الاسم .
٥ - اذا كان رد المستخدم (Yes) ، يتم استخدام الأمر (REPLACE) لاستبدال محتويات الحقل بالاسم الذى تم تخزينه فى متغير الذاكرة .

٦ - يتم تكرار هذه العملية مع الأسماء الأخرى التى يتم ادخالها . وفى كل مرة تستبدل محتويات متغير الذاكرة بالاسم الجديد الذى يتم ادخاله . وهكذا يلاحظ أن متغير الذاكرة (Memory Variable) يعمل كحلقة اتصال بين المستخدم وملف قاعدة البيانات .

ويمكن استخدام متغير الذاكرة أيضا مع المخرجات (Output) التى يتم توجيهها الى الشاشة أو الطابعة بنفس الطريقة التى سبق شرحها .

١٤ - ٤ المتغيرات العامة والمتغيرات الخاصة

يتم تصنيف متغيرات الذاكرة الى متغيرات عامة (Public Variables) ومتغيرات خاصة (Private Variables) . والمتغير العام (Public) هو المتغير الذى يتم انشاؤه فى أى برنامج فرعي أو رئيسى ويكون مؤثرا فى جميع البرامج الأخرى . أما المتغير الخاص (Private) فهو المتغير الذى يتم انشاؤه فى أى برنامج فرعي ولايؤثر الا فى هذا البرنامج أو البرامج الفرعية المتفرعة منه . وفى الأجزاء التالية يتم توضيح خصائص كل من النوعين بشىء من التفصيل .

١٤ - ٤ - ١ المتغيرات العامة (Public Variables)

يتم انشاء المتغير العام فى خطوتين ، الأولى اعلان هذا المتغير كمتغير عام (Public) ، والثانية انشاء هذا المتغير . وذلك كالآتى مثلا :

```
PUBLIC Average
Average = 0.00
```

حيث يتم اعلان (Declare) هذا المتغير كمتغير عام (Public) كما يتم انشاؤه كمتغير عددى (Numeric) يسمح بادخال رقمين عشريين .

وفى هذه الحالة يصبح المتغير (Average) متغيرا عاما . أى أنه يستخدم فى جميع البرامج المستخدمة فى هذا الوقت . سواء كانت برامج رئيسية أو فرعية .

١٤ - ٤ - ٢ المتغيرات الخاصة (Private Variables)

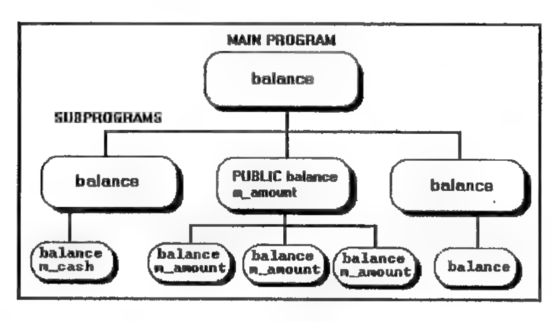
متغير الذاكرة بصفة عامة يصبح خاصا اذا لم يتم اعلانه كمتغير عام (Public) . أى أنه يكفى انشاء المتغير فقط لكى يصبح متغيرا خاصا .

فمثلا الأمر التالى يؤدى الى انشاء متغير خاص فى برنامج فرعى معين :

```
Mcost = 0.00
```

فى هذه الحالة يصبح متغير الذاكرة (Mcost) متغيرا خاصا (Private) فى هذا البرنامج الفرعى والبرامج الفرعية المتفرعة منه .

والشكل (١٤ - ١) يوضح العلاقة بين المتغيرات العامة (Public Variables) و المتغيرات الخاصة (Private Variables) .



شكل (١٤ - ١)

ويلاحظ فى هذا الشكل أن المتغير العام (balance) يؤثر فى جميع البرامج رغم انشائه داخل برنامج فرعى (Subprogram) . أما المتغير (m_amount) فانه يؤثر فقط فى البرامج الفرعية المتفرعة من البرنامج الذى تم انشاؤه خلاله . بينما يعتبر المتغير (m_cash) متغيرا خاصا ببرنامج فرعى واحد ولايؤثر فى باقى البرامج .

ملاحظة

أى متغيرات ذاكرة خاصة يتم انشاؤها فى البرنامج الرئيسى

(Main Program) تؤثر في جميع البرامج الفرعية ، وبالتالى تصبح مثل المتغيرات العامة تماما . ولذلك يفضل دائما انشاء المتغيرات التى يراد استخدامها في جميع البرامج الفرعية مرة واحدة في البرنامج الرئيسى بدلا من الحاجة الى تكرار انشاء متغيرات الذاكرة الخاصة فى كل برنامج فرعى .

١٤ - ٥ التخلص من متغيرات الذاكرة

من المهم جدا التخلص من متغيرات الذاكرة بعد انتهاء البرنامج حتى لاتؤثر فى أى برامج أخرى يراد تشغيلها . كما يراد أحيانا التخلص من بعض هذه المتغيرات أثناء تنفيذ البرنامج . والمتغيرات الخاصة (Private) تختفى بمجرد انتهاء البرنامج أو انتهاء البرنامج الفرعى الذي تم انشاؤها خلاله . وهناك طريقة أخرى للتخلص من متغير ذاكرة خاص أثناء تنفيذ البرنامج وذلك باستخدام الأمر (RELEASE) ، وذلك كالآتى مثلا :

```
RELEASE maverage
```

كما يمكن التخلص من مجموعة من المتغيرات الخاصة بأمر واحد كالآتى :

```
RELEASE ALL LIKE m*
```

ويلاحظ هنا استخدام الحرف الشامل (*) ، وذلك للتخلص من جميع المتغيرات الخاصة التى تبدأ بالحرف (m) .

كما يمكن التخلص من كل المتغيرات الخاصة ماعدا بعض هذه المتغيرات وذلك كالآتى مثلا :

```
RELEASE ALL EXCEPT m*
```

في هذه الحالة يتم التخلص من جميع المتغيرات الخاصة ماعدا المتغيرات التى تبدأ بالحرف (m) .

اما المتغيرات العامة (PUBLIC) فهى لاتختفى باختفاء البرنامج ، بل تظل موجودة في الذاكرة . لذلك يلزم التخلص منها فى نهاية البرنامج باستخدام الأمر (RELEASE) أو (CLEAR MEMORY) أو (CLEAR ALL) .

والأمر (RELEASE) يتم بواسطته التخلص من متغير عام محدد ، ولا تستخدم الحروف الشاملة (Global Characters) في هذه الحالة . ولكن يمكن التخلصمن عدة متغيرات بكتابة أسمائها وبينها فاصلة (Comma) .وذلك كالآتى مثلا :

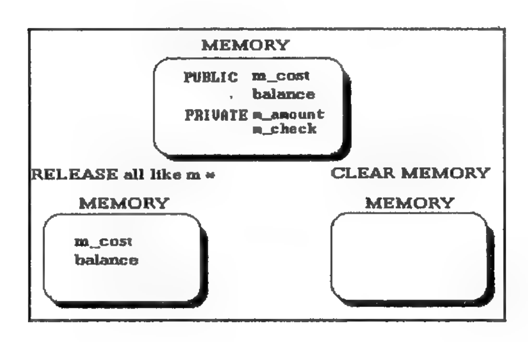
```
RELEASE mcost, mamount
```

أما الأمر (CLEAR ALL) والأمر (CLEAR MEMORY) فيستخدمان للتخلص من جميع المتغيرات العامة التى تم انشاؤها خلال البرنامج .

ملاحظة

المتغيرات العامة بصفة خاصة يلزم التخلص منها خلال البرنامج ، لأنها تظل موجودة فى الذاكرة بعد انتهاء تشغيل البرنامج وقد تؤثر على تشغيل أى برنامج آخر .

والشكل (١٤ - ٢) يوضح كيف يتم التخلص من متغيرات الذاكرة العامة والخاصة .



شكل (١٤ - ٢)

١٤ - ٦ ملفات الذاكرة (Memory Files)

يمكن تخزين متغيرات الذاكرة فى ملف ذاكرة (Memory File) وذلك حتى يمكن استخدام هذه المتغيرات فى أى برنامج دون الحاجة الى انشاء المتغيرات مرة

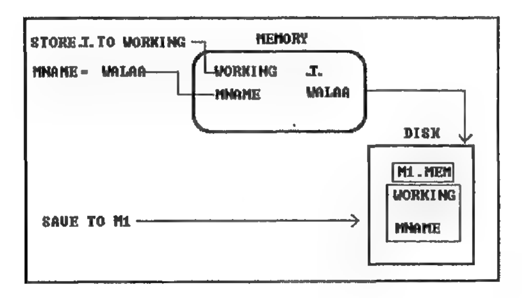
ثانية . ويستخدم الأمر (SAVE) لتخزين محتويات هذه المتغيرات فى ملف الذاكرة . كما يستخدم الأمر (RESTORE) لاسترجاع هذه المتغيرات فى الذاكرة مرة ثانية .

ولانشاء ملف الذاكرة يتم أولا انشاء متغيرات الذاكرة من مشيرة النقطة (Dot Prompt). ثم يتم استخدام الأمر (SAVE) مع اسم الملف المطلوب انشاؤه . ويقوم البرنامج باضافة الامتداد (.mem) الى اسم الملف .

فمثلا عند انشاء عدة متغيرات ذاكرة ويراد تخزينها فى ملف ذاكرة اسمه (M1) ، يستخدم الأمر التالى :

```
SAVE TO M1
```

والشكل (١٤ - ٣) يوضح عملية انشاء ملف ذاكرة



شكل (١٤ - ٣)

١٤ - ٧ استرجاع ملفات الذاكرة

يستخدم الأمر (RESTORE) لاسترجاع ملفات الذاكرة (Memory Files) فى ذاكرة الحاسب المؤقتة (RAM) حتى يمكن استخدام المتغيرات المخزنة فيها وذلك كما يلى :

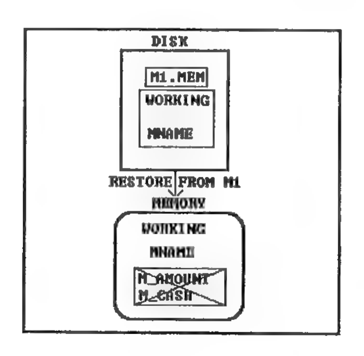
```
RESTORE FROM M1
```

ويجب ملاحظة أن استرجاع أى ملف ذاكرة يؤدى الى مسح كل متغيرات الذاكرة الموجودة فى الذاكرة المؤقتة للحاسب وذلك اذا لم يتم تنبيه البرنامج الى المحافظة على المتغيرات الموجودة فى الذاكرة . ويستخدم الأمر (ADDITIVE) لعمل ذلك . حيث أنه يؤدى الى اضافة المتغيرات الموجودة فى ملف الذاكرة الى المتغيرات الموجودة فى الذاكرة المؤقتة .

فمثلا لكي يتم استرجاع الملف (M1.mem) مع الاحتفاظ بالمتغيرات الموجودة فى الذاكرة المؤقتة يستخدم الأمر التالى :

```
RESTORE FROM M1 ADDITIVE
```

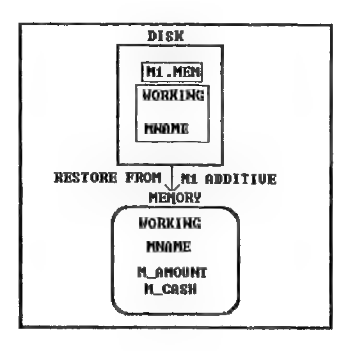
والشكل (١٤ - ٤) يوضح عملية استرجاع ملف الذاكرة باستخدام الأمر (RESTORE) مع فرض وجود متغيرات فى الذاكرة المؤقتة .



شكل (١٤ - ٤)

ويلاحظ اختفاء المتغيرات الموجودة أصلا فى الذاكرة لتحل محلها متغيرات ملف الذاكرة (M1.mem) .

أما الشكل (١٤ - ٥) فيوضح استخدام الأمر (ADDITIVE) لاستعادة نفس الملف .



الشكل (١٤ - ٥)

ويلاحظ فى هذه الحالة اضافة متغيرات ملف الذاكرة الى المتغيرات الموجودة فى الذاكرة المؤقتة .

ملاحظة

كل متغيرات الذاكرة التي يتم انشاؤها من مشيرة النقطة (Dot Prompt) تكون عامة (Public) . ولكن عند استرجاعها بواسطة الأمر (RESTORE) فان حالتها ، أى اذا كانت عامة أو خاصة ، تتوقف على المكان الذى يتم استرجاعها فيه . فاذا استرجعت عند مشيرة النقطة (Dot Prompt) فانها تصبح عامة (Public) ، واذا استرجعت فى البرنامج الرئيسى فانها تصبح عامة أيضا .

أما اذا استرجعت فى برنامج فرعى ، فانها تصبح خاصة (Private) لهذا البرنامج الفرعي والبرامج الفرعية المتفرعة منه فقط . أما اذا أريد استرجاع بعض المتغيرات في برنامج فرعي بحيث تكون متغيرات عامة (Public) ، فيجب اعلانها أولا عامة قبل استرجاع ملف الذاكرة .

فمثلا ملف الذاكرة (Accounts.mem) يحتوى على ثلاثة متغيرات (mcost, mname, mamount). ويراد استرجاع هذا الملف داخل برنامج فرعى بحيث تكون هذا المتغيرات عامة (Public) . فلتنفيذ ذلك يتم كتابة السطرين التاليين :

```
PUBLIC mcost, mname, mamount
RESTORE FROM Accounts
```

فى هذه الحالة تصبح هذه المتغيرات عامة (Public) .

١٤ - ٨ أهمية استخدام ملفات الذاكرة

توفر ملفات الذاكرة (Memory Files) لمخطط البرامج المحترف مرونة عالية فى التحكم فى البرنامج . حيث يمكنه تخزين عدة ملفات ذاكرة كل منها يختص بوظيفة معينة فى البرنامج ، ويمكنه استرجاع الملف المطلوب لأداء هذه الوظيفة فى البرنامج . كما يمكنه استرجاع ملف آخر خاص بوظيفة أخرى ، مع ملاحظة أنه يمكنه التخلص من الملف السابق بمجرد استرجاع الملف الجديد باستخدام الأمر (RESTORE) ، كما يمكنه الاحتفاظ بالملف السابق فى الذاكرة باستخدام الأمر (ADDITIVE) حسب الحاجة . كما يمكنه تعديل محتويات ملف الذاكرة أثناء كتابة البرنامج ، وذلك باضافة متغيرات جديدة أو حذف متغيرات مخزنة به .

فمثلا عندما يحتاج مخطط البرامج الى اضافة متغير ذاكرة جديد الى ملف الذاكرة ، فانه يقوم باسترجاع ملف الذاكرة من مشيرة النقطة (Dot Prompt) ، ثم يتم تخزين المتغير الجديد فى الذاكرة باستخدام الامر (SAVE) . فى هذه الحالة يتم تخزين متغيرات الذاكرة الموجودة فى الذاكرة المؤقتة متضمنة المتغير الذى تمت اضافته .

ملاحظة

ماسبق ذكره فى هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، FoxBase + ، FoxPro.

الباب الخامس عشر

أوامر التجهيز فى البرنامج الرئيسى

البرنامج الرئيسي (Main Program) هو البرنامج الذي يستدعي ويشغل البرامج الفرعية كما يجهز محيط التشغيل (Working Environment) للبرنامج كله، والذي يشمل تحديد وحدة الأقراص المستخدمة أو الفهرس الفرعي المستخدم ، وتحديد حالة الشاشة ، بالاضافة الى وظائف أخرى متعددة سيتم شرحها فيما بعد . كما يقوم البرنامج الرئيسي عادة بعرض القائمة الرئيسية التي يقوم المستخدم بالاختيار منها . كما يقوم البرنامج الرئيسي بفتح ملفات قاعدة البيانات والملفات المصاحبة لها فى بداية البرنامج كما يقوم عادة باغلاق هذه الملفات فى نهاية البرنامج .

١٥ -١ تركيب البرنامج الرئيسى

كما سبق الايضاح فان البرنامج الرئيسي يمكن تقسيمه الى أربعة أقسام وهي المقدمة واوامر التجهيز (Set up) واوامر البرنامج وأوامر الخروج وقد تم فيما سبق توضيح محتويات كل قسم . وفى الأجزاء التالية يتم القاء مزيد من الضوء على أوامر التجهيز (Set up) التى تكتب عادة فى البرنامج الرئيسى لتكون مؤثرة على جميع البرامج المتفرعة منه .

١٥ - ٢ أوامر التجهيز (Set Up)

تبدأ أوامر التجهيز باغلاق جميع ملفات قواعد البيانات ، والتخلص من جميع متغيرات الذاكرة . وذلك لأن هناك احتمالا أن يكون أحد ملفات قواعد البيانات مفتوحا، أو يكون هناك متغيرات ذاكرة موجودة فى الذاكرة المؤقتة (RAM) . ويتم ذلك باستخدام الأمر (CLEAR ALL) .

ويتم بعد ذلك تحديد بيانات محيط التشغيل (Working Invironment) والتى يتم عن طريقها تحديد حالة الشاشة وعمود الحالة (Status Bar) ووحدة الأقراص المستخدمة ... الخ .

١٥ - ٢ - ١ تحديد بيانات محيط التشغيل

عند بداية تشغيل البرنامج يكون هناك وضع مبدئى (Default) لمحيط التشغيل . هذا الوضع المبدئي يشمل مثلا ظهور عمود الحالة (Status bar) أسفل الشاشة بالاضافة الى باقى الرسائل التى تظهر على الشاشة وأشياء أخرى متعددة سيتم دراستها فى هذا الجزء .

وبالنسبة لمخطط البرامج فانه يكون مخيرا بين الاحتفاظ بهذا الوضع المبدئى أو تغييره أو تغيير جزء منه فقط . ويستخدم الأمر (SET) لتنفيذ ذلك مع اضافة المعامل المناسب كما سيتم الايضاح .

١٥ - ٢ - ٢ استخدام الأمر (SET TALK)

الوضع المبدئي لهذا الأمر هو (ON) . وهو يعني ظهور خطوات تشغيل البرنامج على الشاشة أثناء تنفيذه . واذا أراد مخطط البرامج تغيير هذا الوضع فانه يكتب الأمر (SET TALK OFF) في بداية البرنامج .

فمثلا عند كتابة الأمر التالي من مشيرة النقطة (Dot Prompt) يلاحظ ظهور السطر الذي يليه .

```
.STORE "Enter cadet name" TO mname
Enter cadet name
```

ويحدث نفس الشيء بالنسبة لباقي أوامر البرنامج . وهذا يؤدي الى ظهور رسائل على الشاشة ليس مطلوبا ظهورها أثناء تنفيذ البرنامج . ولذلك يتم كتابة الأمر (SET TALK OFF) عادة في بداية البرنامج الرئيسي . وفي هذه الحالة يصبح هذا الأمر مؤثرا على جميع البرامج الفرعية بالاضافة الى البرنامج الرئيسي .

١٥ - ٢ - ٣ استخدام الأمر (SET ESCAPE)

عند الضغط على مفتاح الهروب (Esc) أثناء تشغيل البرنامج ، فان البرنامج يتوقــف . فـاذا أراد مخـطط البرامج أن يمنع حدوث ذلك فانه يكتب الأمر(SET ESCAPE OFF) . وهذا الأمر يؤدي الى ايقاف وظيفة مفتاح الهروب (Esc) أثناء تشغيل البرنامج .

واستخدام هذا الأمر يساعد مخطط البرامج على التحكم في طريقة ووقت الخروج من البرنامج . حيث أن الخروج الفجائي في أي وقت قد يسبب متاعب كثيرة نتيجة عدم التاكد من اغلاق جميع الملفات قبل الخروج .

١٥ - ٢ - ٤ استخدام الجرس (Bell)

الوضع المبدئي للبرنامج هو تشغيل الجرس (Bell) عندما يمتلىء الحقل بالبيانات ، أو عندما يدخل المستخدم مدخلات خطأ . وقد يكون مطلوبا التحكم في هذا الجرس أثناء تشغيل البرنامج ، وذلك بجعله قاصرا على أخطاء معينة للمستخدم .

ويتم الغاء الجرس باستخدام الأمر (SET BELL OFF) . كما يمكن اعادته باستخدم الأمر (SET BELL ON) .

١٥ - ٢ - ٥ استخدام الألوان (Colors)

يمكن التحكم في ألوان الشاشة سواء كانت ألوان الأرضية (Background) أو الأعمدة الضوئية (Highlights) المثلة للحقول أو الكتابة داخل هذه الأعمدة الضوئية . ويتم ذلك باستخدام الأمر (SET COLOR TO) ثم تحديد الألوان المطلوبة لكل منطقة من المناطق التى سبق ذكرها .

كما يمكن تغيير حالة الشاشة من الوان الى أبيض وأسود والعكس . ويتم ذلك باستخدام الأمر (SET COLOR ON/OFF) . حيث يتم التغيير بين (ON) ، (OFF) . كما يمكن استخدام الداله ()ISCOLOR لاختبار الشاشة اذا كانت ملونة أو غير ملونة .وبناء على ذلك يتم التغيير بين الألوان والأبيض والأسود حسب الحاجة .

فمثلا يمكن أن يتضمن البرنامج السطور التالية :

```
IF ISCOLOR ( )
   SET COLOR ON
ENDIF
```

فاذا كانت قيمة الداله ()ISCOLOR صحيحة أى (True) يتم التحويل الى الألوان والعكس صحيح .

١٥ - ٢ - ٦ تعديل وحدة الأقراص المستخدمة

يمكن لمخطط البرامج تعديل وحدة الأقراص المستخدمة .فى أى مكان فى البرنامج حتى يمكن تحميل الملفات الموجودة فى قرص معين ، وذلك باستخدام الأمر (SET DEFAULT TO) ثم كتابة رمز وحدة الأقراص الموجود بها الملفات المراد تحميلها .

فمثلا لاستخدام القرص الصلب (C) يتم كتابة الأمر التالى :

```
SET DEFAULT TO C
```

وفى حالة وجود الملفات فى دليل فرعى (Subdirectory) داخل القرص الصلب مثلا، يتم استخدام الأمر (SET PATH TO) ثم كتابة المسار المطلوب . فمثلا اذا كانت الملفات موجودة فى الدليل الفرعى (C:\cadets) يتم استخدام الأمر التالى :

```
SET PATH TO C:\Cadets
```

كما يمكن الغاء المسار الذى سبق تحديده باستخدام الأمر التالى :

```
SET PATH TO
```

دون تحديد مسار معين .

١٥-٢-٧ اعادة تعريف مفاتيح الوظائف (Function keys)

يسمح البرنامج لمخطط البرامج باعادة تعريف تسع مفاتيح من مفاتيح الوظائف العشرة . حيث أن مفتاح (F1) يكون محجوزا لشاشات المساعدة (Help) التى يستخدمها برنامج (+ DBase III) . وحتى يتم استخدام مفتاح من مفاتيح الوظائف فى تنفيذ أمر معين ، يتم كتابة هذا الأمر بين علامات تنصيص (Quotation) مع كتابة الفاصلة المنقوطة فى نهاية الأمر (حيث أن الفاصلة المنقوطة (;) تمثل مفتاح الادخال) . فمثلا لتخصيص المفتاح (F2) للخروج من البرنامج ، يستخدم الأمر التالى :

```
SET FUNCTION 2 TO 'QUIT;'
```

ويمكن كتابة أى أمر بحيث لايزيد طوله عن ٣٠ حرفا متضمنا الفاصلة المنقوطة .

ملاحظة

يراعى قبل انتهاء كتابة البرنامج اعادة مفاتيح الوظائف الى حالتها الأولى . فمثلا لاعادة المفتاح (F2) الى وظيفته الأولى يستخدم الأمر التالى :

```
SET FUNCTION 2 TO 'ASSIST; '
```

١٥ - ٢ - ٨ التحكم فى عناوين الحقول (Headings)

عند استخدام الأمر (LIST) أو الأمر (DISPLAY) تظهر عناوين الحقول فى السطر الأول ، ويليها البيانات الخاصة بالسجلات المختلفة . وظهور هذه العناوين هو الوضع المبدئى (Default) .

وفى معظم الأحيان يحتاج مخطط البرامج الى وضع عناوين مختلفة لهذه الحقول أو اظهار هذه العناوين بأشكال مختلفة عن الوضع المبدئى ولتنفيذ

ذلك يتم كتابة الأمر التالى :

SET HEADING OFF

وهذا يؤدى الى عدم ظهور عناوين الحقول . ويمكن لمخطط البرامج بعد ذلك كتابة العناوين التى يريدها وبالطريقة التى يريدها كما سيتم الايضاح فيما بعد .

١٥ - ٢ - ٩ اخفاء رسالة المساعدة (Help Message)

عند كتابة أى أمر خطأ من مشيرة النقطة تظهر الرسالة التالية :

Do you want some help (y/n)?

فاذا أراد مخطط البرامج عدم ظهور هذه الرسالة أثناء تنفيذ البرنامج ، فانه يستخدم الأمر التالى :

SET HELP OFF

ويمكن لمخطط البرامج بعد ذلك استخدام رسائل المساعدة التى يريدها كما يمكنه عرض شاشات مساعدة خاصة بالبرنامج توضح للمستخدم مكان الخطأ وطريقة تصحيحه .

كما يمكن اخفاء المستطيل الذى يظهر أعلى الشاشة لتوضيح مفاتيح التصحيح أثناء الكتابة ، وذلك باستخدام الأمر :

SET MENU OFF

كما يمكن بعد ذلك تصميم شاشات التصحيح المناسبة للبرنامج .

١٥ - ٢ - ١٠ الغاء رسالة الأمان (Safety)

عندما يقوم البرنامج بنسخ ملف مكان ملف آخر أو تخزين ملف بعد تعديله تظهر رسالة للمستخدم لتحذيره والتأكد أنه يريد فعلا تنفيذ ذلك . وعادة لايريد مخطط البرامج ظهور هذه الرسائل للمستخدم أثناء تنفيذ البرنامج . ويتم تحقيق ذلك باستخدام الأمر (SET SAFETY OFF) .

١٥ - ٢ - ١١ اخفاء عمود الحالة (Status Bar)

في معظم الأحيان لايريد مخطط البرامج ظهور عمود الحالة أسفل الشاشة . ولتنفيذ ذلك يتم كتابة الأمر (SET STATUS OFF) داخل البرنامج الرئيسي . كما يمكن أيضا اخفاء الرسالة التي تظهر أسفل عمود الحالة ووضع أي رسائل أخرى يريدها مخطط البرامج في نفس المكان عن طريق استخدام الأمر :

```
SET MESSAGE TO
```

ثم كتابة الرسالة المطلوب ظهورها .

١٥ - ٢ - ١٢ اخفاء لوحة الأهداف (Scoreboard)

عند اخفاء عمود الحالة (Status Bar) ، فان برنامج (DBase III +) يستخدم أول سطر أعلى الشاشة في اظهار بعض البيانات التي كانت تظهر على عمود الحالة ، ويسمى هذا السطر لوحة الأهداف (Scoreboard) .

وقد يحتاج مخطط البرامج الى عرض رسائل معينة على هذا السطر ، كما قد يحتاج الى استغلال الشاشة كلها في تصميم شاشة ادخال بيانات . في هذه الحالة فانه لا يريد ظهور أي رسائل فجائية على هذا السطر . ويتم ذلك باستخدام الأمر التالي :

```
SET SCOREBOARD OFF
```

وهناك العديد من أوامر التجهيز الأخرى التي تكتب في البرنامج الرئيسي والتي يستخدم فيها الأمر (SET) ، وسيتم دراستها بالتفصيل في الجزء الخاص بالأوامر (Commands) في الكتاب الثاني .

ملاحظة

عادة يتم كتابة مجموعة أوامر التجهيز التي سبق شرحها في بداية أي برنامج رئيسي (Main Program) . ويمكن توفيرا للجهد كتابة هذه الأوامر في ملف أوامر (Command File) منفصل ونسخها في أي برنامج جديد يراد تصميمه .

ملاحظة

ماسبق ذكره في هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، FoxBase + ، FoxPro .

الباب السادس عشر

التحكم فى الشاشة من خلال البرنامج

يعتبر تصميم شاشات ادخال البيانات (Custom Screens) من أهم وسائل تحقيق التفاعل بين البرنامج والمستخدم . فكلما كانت هذه الشاشات واضحة للمستخدم وقريبة الى نماذج البيانات التى يستخدمها ، كلما كان من السهل عليه تشغيل البرنامج والاستفادة منه . كما أن عرض الرسائل والارشادات الواضحة للمستخدم على الشاشة بطريقة واضحة ومفهومة يؤدى الى سهولة متابعة البرنامج ، والحصول على أكبر كفاءة له .

والبرنامج يتيح لمخطط البرامج مرونة كبيرة فى تصميم شاشات ادخال البيانات وعرض أى رسائل للمستخدم بما يحقق التفاعل بينه وبين البرنامج .

١٦ - ١ احداثيات الشاشة

تنقسم شاشة الحاسب الى ٢٥ سطرا أفقيا ، و٨٠ عمودا رأسيا ومن تقاطع هذه الصفوف مع الأعمدة تتحدد النقط التى يمكن كتابة حروف فيها . ويتم ترقيم الصفوف من أعلى الى اسفل بدءا من الصفر (0) وانتهاء بأربعة وعشرين (24) . أما الأعمدة فترقم من اليسار الى اليمين بدءا من الصفر (0) وانتهاء بتسعة وسبعين (79) . ويتم تحديد موقع أى نقطة على الشاشة بكتابة رقم الصف (Row) أولا ثم رقم العمود (Column) .

فمثلا النقطة (1,5) هى النقطة الناتجة عن تقاطع الصف رقم واحد مع العمود رقم ٥ . وهكذا .

ملاحظة

عند استخدام الأمر (SET STATUS ON) يصبح السطر رقم صفر أعلى الشاشة هو لوحة الأهداف (Scoreboard) . فعندما يراد استخدام السطر الأول فى عرض أى بيانات على الشاشة من خلال البرنامج مع تلافى أى مشاكل تنتج عن عرض رسائل فجائية للبرنامج فى هذا السطر ، يتم استخدام الأمر (SET SCOREBOARD OFF).

١٦ - ٢ استخدام الأمر (@ ... SAY)

يستخدم هذا الأمر فى تصميم شاشات ادخال البيانات . كما يستخدم فى عرض أى بيانات أو رسائل على الشاشة . فمثلا لعرض محتويات حقل الاسم (Name) لسجل معين بدءا من السطر الرابع والعمود التاسع (٤ ، ٩) يتم استخدام الأمر التالى :

```
@ 4,9 SAY Name
```

و حرف (@) هنا يستخدم بمعنى عند او "AT" و ذلك لتحديد موقع بدء الكتابة.

وعندما يراد عرض بيانات متغير ذاكرة (Memory Variable) ، يجب التأكد أولا من انشاء هذا المتغير . وذلك كالآتى مثلا :

```
STORE "Enter New Name" To message
@ 10,15 SAY message
```

١٦ - ٣ مسح الشاشة

عند عرض بيانات على الشاشة يجب أولا مسح البيانات السابقة الموجودة على الشاشة . ويتم ذلك باستخدام الأمر (CLEAR) . ويمكن مسح جزء فقط من الشاشة باستخدام الأمر (CLEAR ... @) مع كتابة الاحداثيات المطلوب بدء المسح عندها بعد الحرف @ .

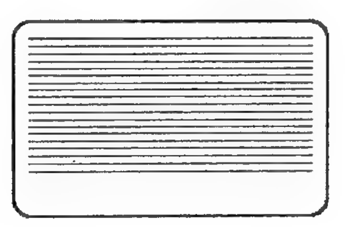
فمثلا عندما يراد مسح الشاشة ابتداء من السطر ١٩ الى آخر الصفحة مع ترك السطور من صفر الى ١٨ على الشاشة ، يتم استخدام الأمر التالى :

```
@ 19,0 CLEAR
```

انظر الشكل (١٦ - ١)

ملاحظة

يجب ملاحظة الفرق بين الأمر (CLEAR) المستخدم فى مسح الشاشة والأوامر (CLEAR ALL) أو (CLEAR MEMORY) المستخدمة فى مسح متغيرات الذاكرة .

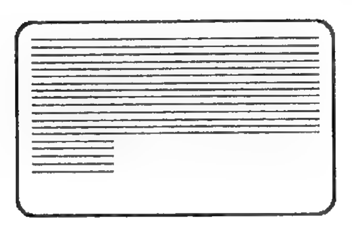


شكل (١٦ - ١)

كما يمكن مسح الشاشة ابتداء من سطر معين وعمود معين كالآتي مثلا :

```
@ 15,15 CLEAR
```

انظر الشكل (١٦ - ٢)



شكل (١٦ - ٢)

كما يمكن مسح سطر واحد عن طريق كتابة احداثيي هذا السطر مع عدم كتابة أى شيء فيه . وذلك كالآتي مثلا :

```
@ 15,0
```

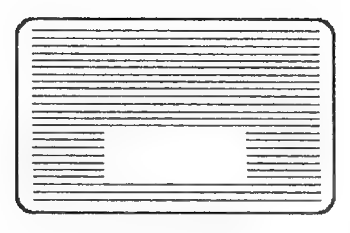
كما يمكن مسح مجموعة من السطور مسحا تدريجيا ، أى سطرا سطرا كالآتي مثلا :

```
@    19,0
@    20,0
@    21,0
@    22,0
@    23,0
@    24,0
```

كما يمكن مسح مساحة مستطيلة أو مربعة من الشاشة عن طريق تحديد احداثي النقطة أعلى يسار هذا المستطيل ، والنقطة أسفل يمين هذا المستطيل . وذلك كالآتي مثلا :

```
@ 15,15 CLEAR TO 21,50
```

أنظر الشكل (١٦ - ٢)

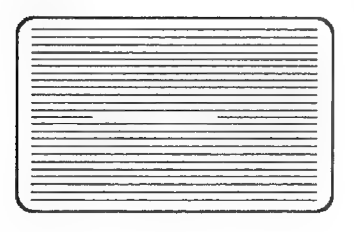


شكل (١٦ - ٣)

كما يمكن مسح جزء من السطر فقط عن طريق كتابة مسافات خالية (Spaces) فى هذا السطر . وذلك كالآتى مثلا :

```
@ 13,15 SAY SPACE(20)
```

انظر الشكل (١٦ - ٤)



شكل (١٦ - ٤)

مما سبق يلاحظ أن البرنامج يتيح لمخطط البرامج أكبر مرونة ممكنة فى التحكم فى الشاشة وعرض ومسح أى رسائل أو بيانات فى أى مكان منها .

١٦ - ٤ عرض نص على الشاشة

كما سبق الايضاح فان الأمر (SAY . . . @) يستخدم فى عرض أى رسائل (Messages) أو بيانات على الشاشة . وعندما يراد عرض نص كبير (Text) على الشاشة يمكن استخدام الأمر (SAY ... @) أيضا فى كتابة كل سطر فى النص على سطر محدد في الشاشة . ولكن هذه الطريقة قد تبدو طويلة وتستهلك كثيرا من الوقت خصوصا فى النصوص الكبيرة التي يراد عرضها من خلال البرنامج مثل شاشات المساعدة (Help Screens) . لذلك فالبرنامج يتيح وسيلة أخرى أسرع فى كتابة النصوص الكبيرة .وذلك باستخدام الأمر (TEXT....ENDTEXT) . حيث يتم كتابة (TEXT) في أول سطر قبل النص المراد كتابته ، ثم كتابة النص المطلوب ثم كتابة (ENDTEXT) بعد آخر سطر فى النص .

ملاحظة

يجب ملاحظة أن الأمر (TEXT ... ENDTEXT) لايستخدم السطر رقم صفر فى كتابة النص . كما أنه لايستخدم في عرض بيانات الحقول أو متغيرات الذاكرة .

١٦ - ٥ استخدام الأمر (@ ... GET ... READ)

يستخدم هذا الأمر فى ادخال المدخلات التى يقوم المستخدم بادخالها . والجزء الأول من الأمر وهو (GET ... @) يؤدى الى عرض عمود ضوئى (Highlight)

بدءا من النقطة التى تكتب احداثياتها بعد الحرف @ . كما يستخدم الجزء الآخر (READ) في تخزين البيانات التى يكتبها المستخدم فى هذا العمود الضوئى فى المتغير الذى يتم كتابة اسمه بعد الأمر (GET) . وذلك كالآتى مثلا:

```
@ 15,15 GET mname READ
```

ويجب التاكد أن متغير الذاكرة (mname) قد سبق انشاؤه .

ويستخدم هذا الأمر فى تصميم شاشات ادخال بيانات مثل الشاشات التى تظهر عند استخدام الأمر (APPEND) أو الأمر (EDIT) . حيث يتم كتابة هذا الأمر مع متغيرات الذاكرة التى يتم انشاؤها ممثلة لحقول البيانات فى ملف قاعدة البيانات ، على أن يتم تحديد مكان العمود الضوئى (Highlight) الممثل لهذا الحقل عن طريق الاحداثيات التى تكتب بعد الحرف @ . كما يتم كتابة رسائل للمستخدم قبل العمود الضوئى لتوضيح البيان المطلوب ادخاله وذلك باستخدام الأمر(SAY ... @) .

فمثلا اذا كان هناك ثلاثة حقول فى ملف قاعدة البيانات للاسم والعنوان ورقم التليفون ، وهي (tel ، address ، name) على الترتيب . فى هذه الحالة يجب أولا انشاء متغيرات الذاكره (mtel ، maddress ، mname) على الترتيب مثلا ، ثم كتابة الأوامر التى تؤدى الى ظهور الشاشة المطلوبة وذلك كالآتى :

```
mname    =  SPACE(20)
maddress=SPACE(20)
mtel    =  SPACE(10)
CLEAR
@     3,2   SAY 'Enter Name:'
@     3,30 GET mname
@     5,2   SAY 'Enter Address:'
@     5,30 GET 'maddress'
@     7,2   SAY 'Enter Telephone No:'
@     7,30 GET mtel
READ
```

وعند تنفيذ البرنامج تظهر الشاشة المبينة فى الشكل (١٦ - ٥).

```
Enter Name :[                    ]
Enter Address :[                 ]
Enter Telephone No: [            ]
```

شكل (١٦ - ٥)

ويلاحظ فى هذا المثال استخدام أمر (READ) واحد لجميع السطور . وهذا يؤدى الى ظهور جميع الأعمدة الضوئية (Highlights) مرة واحدة على الشاشة . كما يؤدي الى ظهور مؤشر صغير فى آخر عمود ضوئي (Highlight) يساعد على تصحيح أي بيانات يقوم المستخدم بادخالها خطأ . كما يمكن نقل المؤشر الى أى عمود ضوئي آخر وتصحيح البيانات التى سبق ادخالها . وذلك باستخدام مفاتيح الأسهم (<-- , --> , ↑ , ↓) .

وهناك طريقة أخرى لاستخدام الأمر (READ) ، وذلك بكتابته بعد كل أمر (GET). وهذا يؤدى الى ظهور أول عمود ضوئي فقط والانتظار حتى يكتب المستخدم بيانات في هذا العمود ثم يضغط على مفتاح الادخال ، فيظهر العمود التالى وينتظر حتى يكتب المستخدم فيه ويضغط على مفتاح الادخال فيظهر العمود الثالث . .. وهكذا . ويمكن استخدام هذه الطريقة مع نفس المثال السابق كالآتى :

```
CLEAR
@     3,2  SAY 'Enter Name:'
@     3,30 GET mname
READ
@     5,2  SAY 'Enter Address:'
@     5,30 GET 'maddress'
READ
@     7,2  SAY 'Enter Telephone No:'
@     7,30 GET mtel
READ
```

وفى هذه الحالة لايستطيع المستخدم الرجوع الى الحقول السابقة لاصلاح البيانات التى قام بادخالها فيها الا باستخدام الأمر (READ SAVE) بدلا من الأمر (READ). حيث أن الأمر (READ SAVE) يمكن المستخدم من الرجوع الى الحقول السابقة عن طريق مفاتيح الأسهم(--> , <-- , ↑ ↓) وتصحيح البيانات الموجودة فيها .

ويمكن عرض عدة أعمدة ضوئية (Highlights) على سطر واحد . وذلك عن طريق تثبيت رقم السطر وتغيير رقم العمود . ويمكن ملاحظة ذلك فى السطور التالية :

```
@     5,0  GET mname
@     5,30 GET maddress
@     5,60 get mtel
READ
```

وعند تنفيذ هذه السطور يظهر الآتى على الشاشة :

ملاحـظة

يجب ملاحظة أن عدد أماكن التخزين المسموح بها للأمر (GET) لايزيد عن ١٢٨ مكانا . وهو يعتبر عددا كافيا جدا لأن كل مكان من هذه الأماكن يتم مسحه آليا بمجرد استخدام الأمر (READ) فى تخزين البيانات الموجودة فيه . ولكن هذا المكان يظل محجوزا فى حالة استخدام الأمر (READ SAVE) . وفى هذه الحـالة يلزم اسـتخدام الأمـر (CLEAR GETS) لمسح الأماكن المحجوزة والاحتفاظ بعدد كبير من الـ (GETS) التى يمكن استخدامها فى ادخال مدخلات جديدة .

١٦ - ٦ انشاء شاشة مكونة من عدة صفحات

عندما يكون عدد حقول قاعدة البيانات كبيرا ، ولايمكن عرضه على شاشة واحدة، فان البرنامج يتيح لمخطط البرامج تصميم شاشة ادخال مكونة من عدة صفحات ، وذلك باستخدام الأمر (CLEAR) لمسح كل شاشة والانتقال الى الشاشة التالية . ويمكن فى هذه الحالة استخدام أمر (READ) مرة واحدة بعد أوامر (GET) ، مع ملاحظة استخدام الأمر (GLEAR GETS) عندما يزيد عدد الـ (GETS) عن ١٢٨ .

ملاحـظة

عند ادخال البيانات فى العمود الضوئى (Highlight) ينتقل المـؤشر الى العمود الضوئى التالى فى حالة امتلاء العمود الضوئى الأول . أما فى حالة عدم امتلاء العمود الأول بالبيانات فيجب فى هذه الحالة الضغط على مفتاح الادخال حتى ينتقل المـؤشر الى العمود التالى . وهذا هو نفس ما يحدث عند استخدام الأمر (APPEND) أو الأمر (EDIT) .

ويمكن تغيير هذا الوضع ، أى عدم انتقال المؤشر الى العمود التالى الا بالضغط على مفتاح الادخال بصرف النظر عن امتلاء العمود الضوئى أو عدم امتلائه ، وذلك باستخدام الأمر (SET CONFIRM ON) .

١٦ - ٧ استخدام الأمر (ACCEPT) والأمر (INPUT)

كما سبق الايضاح فان الأمر (@...GET) يستخدم عادة عندما يراد استقبال المدخلات من المستخدم فى عدة متغيرات ذاكرة . أما اذا أريد الادخال فى متغير ذاكرة واحد فقط ، مثل الاجابة على سؤال معين وتخزين هذه الاجابة فى متغير ذاكرة، فى هذه الحالة يفضل استخدام الأمر (ACCEPT) أو الأمر (INPUT) حيث أن استخدامهما يكون أسرع وأسهل .

والأمر (ACCEPT) والأمر (INPUT) متشابهان في الوظيفة . فهما يطلبان من المستخدم كتابة شيء معين ، ثم يخزنان ما يكتبه المستخدم فى متغير ذاكرة . هذا المتغير يمكنه تخزين حتى ٢٥٤ حرفا .

وهما يختلفان عن الأمر (GET ... @) فى أنهما لايتطلبان انشاء متغير الذاكرة أولا ، لأنهما يقومان بهذه العملية آليا . كما أنهما لايعرضان عمودا ضوئيا (Highlight) للكتابة فيه .

والفرق الوحيد بين الأمرين (ACCEPT) و (INPUT) هو أن الأمر (ACCEPT) يقبل مدخلات حرفية (Character) فقط . حتى اذا أدخل المستخدم أعدادا فانه يعامل هذه الأعداد كحروف .

أما الأمر (INPUT) فانه يقبل أى نوع من المدخلات سواء كانت حرفية (Character) أو عددية (Numeric) أو تاريخية (Date) أو منطقية (Logical) ، ثم يقوم بانشاء متغير الذاكرة المقابل . وعند ادخال مدخلات حرفية يشترط وضعها بين علامات تنصيص (Quotation Marks) . وهذا عكس الأمر (ACCEPT) الذى لايتطلب وضع علامات التنصيص .

وعند كتابة أى أمر من هذين الأمرين ، يتم كتابة رسالة للمستخدم أولا ثم تحديد اسم متغير الذاكرة الذى يتم ادخال ما يكتبه المستخدم فيه . وذلك كالآتى مثلا :

```
ACCEPT "What is your name?" TO mname
```

وهذا الأمر يطلب من المستخدم كتابة اسمه . ثم يخزن هذا الاسم فى متغير الذاكرة (mname) .

كما يمكن كتابة الأمر التالى :

```
INPUT "Enter your age" TO mage
```

وهذا الأمر يطلب من المستخدم كتابة قيمة عددية تمثل عمره . ثم يخزن هذه القيمة العددية فى المتغير (mage) .

١٦ - ٨ استخدام الأمر (WAIT)

يستخدم الأمر (WAIT) لعمل توقف مؤقت للبرنامج (Pause) حتى يقوم المستخدم بالضغط على أى مفتاح لاستكمال تنفيذ البرنامج . وذلك حتى يعطى المستخدم الفرصة لقراءة رسائل معينة أو بيانات على الشاشة . ويفضل استخدام الأمر

(SET ESCAPE OFF) كما سبق الايضاح حتى لايتسبب ضغط المستخدم على مفتاح الهروب (Esc) فى توقف البرنامج تماما .

والأمر (WAIT) يؤدى الى عرض الرسالة التالية :

```
Press any key to continue
```

وذلك عند كتابته دون كتابة أى رسالة معه كالآتى مثلا :

```
WAIT
```

ويمكن لمخطط البرامج كتابة أى رسالة يريد عرضها للمستخدم ..وذلك كالآتى مثلا :

```
WAIT "Press any key to return to main menu"
```

وهناك فرق واضح بين الأمر (WAIT) والأمرين (ACCEPT) ، (INPUT) . حيث أن الأمر (WAIT) لايخزن ما يكتبه المستخدم في متغير ذاكرة (Memory Variable). وانما يقوم بايقاف تنفيذ البرنامج ايقافا مؤقتا حتى يضغط المستخدم على أى مفتاح للاستمرار . ولكن مع ذلك فيمكن في بعض الحالات تخزين الحرف الذى يقوم المستخدم بادخاله في متغير ذاكرة عندما يتطلب البرنامج ذلك . و يمكن ملاحظة ذلك من مجموعة الاوامر التالية :

```
WAIT "press R to return to mainmenu , or " + ;
" any other  key to continue " TO mmenu
IF UPPER (mmenu) = "R"
    RETURN
ELSE
    -------
    -------
    -------    commands
    -------
    -------
ENDIF
```

هذه الاوامر تؤدى الى توقف تنفيذ البرنامج حتى يضغط المستخدم على اى حرف . فاذا ضغط على الحرف (R) يتم الرجوع الى القائمة الرئيسية . واذا ضغط على اى مفتاح آخر يتم استمرار تنفيذ اوامر البرنامج . و يلاحظ هنا استخدام علامة (;) لامتلاء السطر الاول و اكمال كتابة الامر فى السطر التالى .

ملاحظة

لا يمكن استخدام احداثيات الشاشة في تحديد مكان ظهور الرسالة الخاصة بالاوامر الثلاثة (ACCEPT) ، (INPUT) ، (WAIT) .

ولكن يمكن التحكم في اماكن هذه الرسائل عن طريق اضافة سطور خالية بعد آخر رسالة معروضة على الشاشة . و يتم ذلك باستخدام الامر (?) كما سيتم الايضاح فيما بعد .

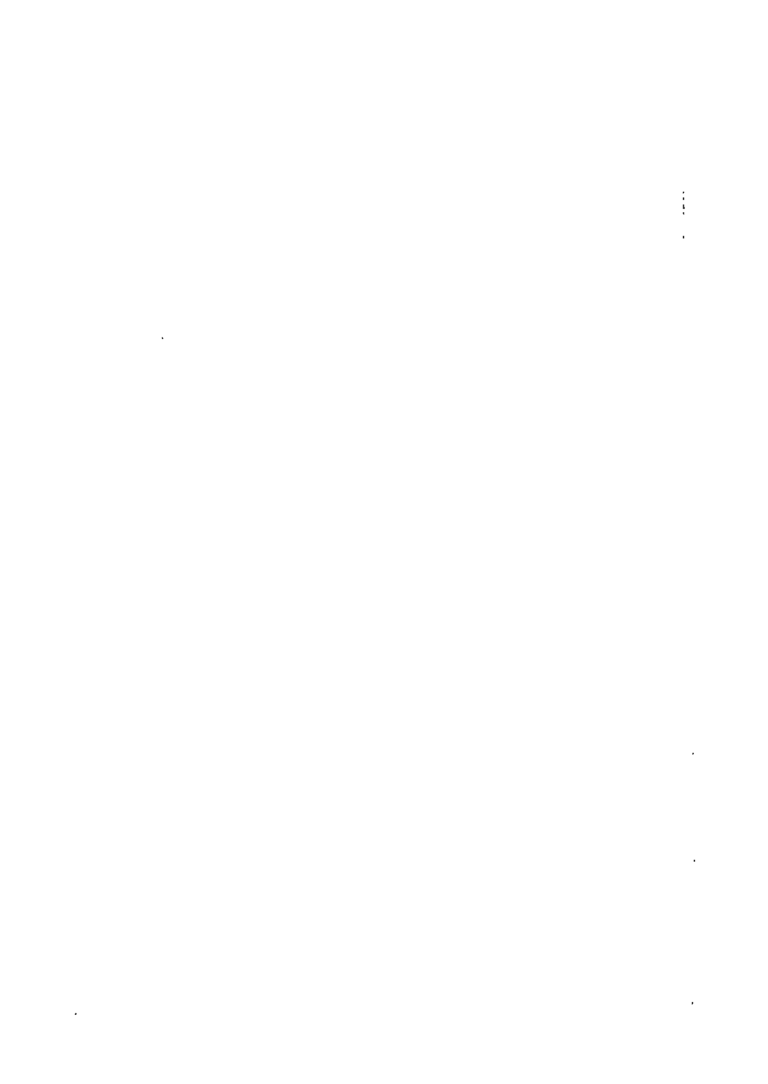
ملاحظة

ماسبق ذكره فـــي هذا البـــاب ينطبق أيضـــا عـــلى كـــل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، + FoxBase ، FoxPro .

الباب السابع عشر

التحكم فى شكل ومدى المدخلات

Templates and Ranges



من المهم جدا ان يحتوى تصميم البرنامج على امكانيات التحكم فى شكل المدخلات (Template) ومدي هذه المدخلات (Range) . لان هذا التحكم يؤدى الى تأمين قاعدة البيانات ضد أى مدخلات خطأ يقوم المستخدم بادخالها. فمثلا عندما يكون مطلوبا ادخال قيمة عددية فى حقل معين ، فان تحديد الشكل (Template) يؤدى الى قبول مدخلات عددية ولا يقبل أى مدخلات أخرى . كما أن تحديد مدى معين (Range) لهذه الأعداد يؤدى الى تأمين البرنامج ضد ادخال أى قيم قد تؤدى الى توقف البرنامج .

ولتحديد شكل المدخلات يستخدم التعبير (PICTURE) ، ولتحديد مدى المدخلات يستخدم التعبير (RANGE) . و سيتم دراسة هذين التعبيرين بالتفصيل فى الأجزاء التالية .

١٧ - ١ استخدام التعبير (PICTURE)

يستخدم التعبير (PICTURE) كمكمل للأمر (@...SAY) والأمر (@...Get) . و يلى هذا التعبير خليط من واحد أو أكثر من الرموز الممثلة للشكل (Template Symbols)و الدوال (Functions) .

وعند استخدام هذا التعبير (PICTURE) مع الأمر (@...Say) فانه لايغير البيانات المخزنة فى الحقول أو متغيرات الذاكرة ولكنه يغير شكل هذه البيانات عند ظهورها على الشاشة .

اما استخدامه مع الأمر (@...GET) فانه يؤدى الى التحكم فى البيانات التى يقوم المستخدم بادخالها قبل تخزينها فى الحقول أو متغيرات الذاكرة.

١٧ - ٢ استخدام رموز الشكل (Template Symbols)

تستخدم رموز الشكل فى تحديد نوع الحروف التى يتم ادخالها . كما يمثل عدد رموز الشكل عدد الحروف التى يتم كتابتها . فمثلا يمكن ملاحظة المثال التالى :

```
@ 15,15 SAY mprice Picture "999999.99 "
```

والرقم (9) هنا يمثل رمز يوجه البرنامج الى اظهار أعداد فقط كما يمثل عدد أرقام (9) عدد الأرقام التى يسمح البرنامج بظهورها . كما تؤدى النقطة (.) الى تحديد عدد الكسور العشرية .

والوضع المبدئي لظهور الاعداد عند استخدام الامر(GET... @) هو ظهور عشرة أرقام . فاذا أريد عرض أو ادخال أعداد أكبر من ذلك يتم كتابة عددا من الرموز يمثل هذه الأرقام . فمثلا يمكن ادخال الأوامر التالية :

```
mamount = 0
@ 15,15 GET mamount PICTURE"999999999999"
```

وهذه الأوامر تؤدى الى الحصول على عدد مكون من ١٢ رقما .

ويلاحظ هنا ضرورة انشاء متغير الذاكرة (mamount) قبل استخدام الأمر (@...GET) كما سبق الايضاح . وفى هذه الحالة يتم اعطاؤه القيمة صفر حيث أن أى قيمة عددية تكون كافية لانشاء المتغير .

ومن الرموز التى تستخدم فى أغلب البرامج الرمز (!) . وهذا يؤدى الى تحويل الحروف التى يدخلها المستخدم الى حروف كبيرة (Uppercase) . مثلا يمكن ادخال الأوامر التالية :

```
mchoice = " "
@ 10,5 SAY "Enter your choice A,B,C,D"
@ 10,50 GET mchoice PICTURE "!"
READ
```

وتؤدى هذه الأوامر الى ظهور رسالة للمستخدم تطلب منه ادخال حرف معين من الحروف (A,B,C,D) . وعند كتابة المستخدم لهذا الحرف سواء كتبه صغيرا أو كبيرا يتم تحويله الى حرف كبير (Uppercase) ثم تخزينه فى المتغير (mchoice).

و يلاحظ هنا أنه تم كتابة رمز واحد لأن المتغير يحتوى على حرف واحد فقط . أما اذا أريد ادخال عدد معين من الحروف ، فيتم كتابة عدد من الرموز يماثل عدد الحروف المطلوب ادخالها . فمثلا يمكن كتابة الأمر التالى :

```
@ 15,15 GET mname PICTURE "AAAAAA "
```

ويؤدى هذا الأمر الى ادخال اسم يحتوى على ستة حروف . والحرف (A) هنا يعنى قبول الحروف الأبجدية فقط ، ولايقبل الأعداد أو الحروف الخاصة (Special characters).

و عندما يراد ادخال بيانات حرفية تحتوى على أعداد أو حروف خاصة يتم

استخدام الرمز (X) . و الأمر التالى يوضح ذلك :

```
@ 10,10 GET maddr PICTURE "XXXXXXXXXXXXXXXXXXXX "
```

هذا الأمر يعني قبول ٢٠ حرفا تمثل العنوان (maddr) وتحتوى على أى نوع من الحروف سواء كانت أرقاما أو حروفا خاصة .

ويمكن التحكم فى شكل الاعداد التي تظهر على الشاشة باستخدام عدة رموز من رموز الشكل (Template Symbols) . فمثلا علامة ($) التى تسمى علامة الدولار (Dollar Sign) يتم كتابتها بعد التعبير (PICTURE) . وهى تؤدى الى ظهور هذه العلامة مكان أى أصفار يسار العدد كما يتضح من الأمر التالى :

```
@ 10,10 SAY mamount PICTURE  "$$$$$$.$$"
```

فاذا كان المتغير (mamount) يحتوى على العدد (66) مثلا يظهر العدد على الشاشة كالآتى :

```
$$$$66.00
```

و اذا أريد اظهار علامة دولار واحدة قبل العدد يتم استخدام الدالة (STR) التى سيتم شرحها فيما بعد . وذلك يتضح من الأمر التالى :

```
@ 10,10 SAY "$" + STR (mamount, 5,2)
```

ويؤدى هذا الأمر الى اظهار العدد على الصورة التالية :

```
$ 66.00
```

ويلاحظ هنا استخدام علامة الجمع (+) فى جمع البيانات الحرفية (Concatination) . و هذا سيتم شرحه فيما بعد.

كما يمكن استخدام الفاصلة (Comma) داخل العدد كما يتضح من السطور التالية :

```
mprice = 0.00
@ 10,10 GET mprice PICTURE "999,999.99"
READ
```

ومن الرموز التي تستخدم في المدخلات المنطقية الرمز (Y) وهو يحدد الحرف المنطقي الذي يدخله المستخدم ليكون (Y) وتعنى نعم (Yes) او يكون (N)وتعنى لا (NO) .

و يتضح ذلك من السطور التالية :

```
@ 10,10 SAY "Send reports to printer (Y/N) " ;
GET mprint PICTURE "Y"
READ
```

١٧ - ٢ استخدام دوال الشكل (Template Functions)

دوال الشكل هي دوال تتحكم في شكل المدخلات كلها وليس فى حرف واحد ، أي تستخدم دالة واحدة فقط فى التحكم في شكل الحروف كلها. ويمكن استخدام الدوال مع رموز الشكل في أمر واحد. ولكن يجب أن تكون الدالة فى أول العبارة التى تلى التعبير (PICTURE) ويتم فصلها عن الرموز بمسافة واحدة (Space). وتبدأ الدالة عادة بالحرف (@) ثم الحرف الذى يمثل هذه الدالة . فمثلا الأمر التالى يوضح احدى هذه الدوال .

```
@ 5,5 SAY mcost PICTURE "@B 9,999,999/99 "
```

وهذا الأمر يؤدى الى عرض الرقم مضبوطا من اليسار (Left Justified) بدلا من عرضه مضبوطا من اليمين كما يحدث فى العادة . كما يؤدى هذا الأمر ايضا الى عرض الأعداد وبها فواصل (Commas) بعد كل ثلاثة أرقام .

ويمكن استخدام كلمة (Function) بدلا من الحرف (@) ولكنها فى هذه الحالة تكتب قبل التعبير (PICTURE) كما فى الأمر التالى :

```
@ 5,5 SAY mcost FUNCTION "B" PICTURE "9,999,999.99 "
```

كما يمكن جمع عدة دوال فى أمر واحد فمثلا فى برامج المحاسبة عندما يراد عرض نوع الرصيد اذا كان دائنا (Credit) و مدينا (Debit) بدلا من علامة السالب والموجب على الترتيب يستخدم الأمر التالى :

```
@ 5,5 SAY mamount PICTURE "@XC 999,99 "
```

حيث تؤدى الدالة (X) الى اظهار الحروف (DB) اى مدين (Debit) بعد

العدد السالب . كما تؤدى الدالة (C) الى اظهار الحروف (CR) اى دائن (Credit) بعد العدد الموجب .

ويمكن استخدام الدالة (Z) التى تمسح الأصفار الموجودة يسار الرقم . أو أى أصفار أخرى ليس لها قيمة .

كما تستخدم الدالة (E) فى تحويل التاريخ (Date) الى الشكل الأوربى (dd/mm/yy) . والدالة (D) فى تحويل التاريخ الى الشكل الأمريكى (mm/dd/yy) . و الأوامر التالية توضح ذلك :

```
mdate = CTOD ('03/10/89')
@ 5,5 SAY mdate PICTURE '@E'
```

وهذه الأوامر تؤدى الى عرض التاريخ (mdate) على الصورة (10/03/89) .

ويلاحظ من الأمر فى السطر الأول استخدام الدالة (CTOD) وهى اختصار (Character To Date Conversion Function) لتحويل التاريخ من الحروف (Characters) الى تاريخ (Date) حتى يتم تخزينه فى المتغير التاريخى (mdate) .

كما تستخدم الدالة (R) فى ادخال أى حروف داخل العبارة الخاصة بالتعبير (PICTURE) بحيث تستخدم هذه الحروف كفواصل للحروف التى يدخلها المستخدم . فمثلا عندما يراد عرض الحروف وبينها مسافات يمكن كتابة الأمر التالى :

```
@ 5,5 SAY mname PICTURE "@R X X X X X X "
```

فى هذه الحالة تظل المسافات كما هى وتظهر الحروف مكان حروف (X) . فاذا كان المتغير (mname) يحتوى على الاسم (AHMED) مثلا يظهر الاسم على الشاشة كالآتى :

```
A H M E D
```

ويجب ملاحظة أن هذه الحروف التى يتم اضافتها كفواصل لاتخزن بعد ذلك فى المتغير عند استخدام الأمر (GET) مثلا، ولكنها تظهر على الشاشة فقط.

كما تستخدم الدالة (S) فى عمل ازاحة أفقية (Horizontal Scrolling) فى العمود الضوئى (Highlight) الذى يظهر مع الأمر (GET ... @) . و يفيد هذا عندما يكون هناك مكان محدود للكتابة

فيه ويراد كتابة بيانات تزيد عن العمود الضوئى . حيث تسمح هذه الازاحة باضافة حروف جديدة دون زيادة طول العمود الضوئى . ويتم ذلك عن طريق كتابة رقم بعد الدالة (S) يحدد طول العمود الضوئى المراد عرضه . كما يتم اضافة رموز للشكل (Template Symbols) تمثل أقصى عدد من الحروف المطلوب ادخالها . ويتضح ذلك من مجموعة الأوامر التالية :

```
mname = SPACE(20)
@ 10,10 "Enter a name" ;
GET mname PICTURE "@ S8 !!!!!!!!!!!!!!!!!!!!"
READ
```

Ahmed Za

قبل الازاحة

med Zaky

بعد الازاحة

فعندما يتم كتابة اسم يزيد عن طول العمود الضوئى ، يلاحظ تحرك الحروف جهة اليسار حتى تسمح بادخال الحروف الجديدة .

وهناك عدة دوال شكل (Template Functions) ورموز شكل (Template Symbols) يمكن استخدامها فى التحكم فى شكل المدخلات . ولكن لامجال لشرحها بالتفصيل فى هذا الجزء .

ارجع الى الجزء الخاص بالدوال (Functions) فى الكتاب الثانى .

١٧ - ٤ تحديد المدى (RANGE)

يستخدم التعبير (RANGE) لتحديد المدى بالنسبة للمدخلات العددية أو التاريخية . حيث يستخدم مع الأمر (@...GET) لتحديد أصغر وأكبر قيمة يمكن للمستخدم ادخالها . ويجب ملاحظة أن التواريخ (Dates) هى فى الحقيقة أعداد ، حيث أن كل تاريخ يمثل داخل قاعدة البيانات بعدد معين .

فمثلا عندما يراد ادخال عدد فى المتغير (mprice) يكون محصورا بين عشرة وألف ، يتم كتابة الأمر التالى :

```
@ 5,5 GET mprice RANGE 10,1000
```

اما اذا أريد تحديد قيمة المتغير بحيث لا تقل عن (10) ، فى حين يمكن أن تزيد الى أى عدد ، يتم كتابة الأمر التالى :

```
@ 5,5 GET mprice RANGE ,10
```

وبالنسبة للمتغيرات التاريخية يجب استخدام الدالة (CTOD) فى تحويل التاريخ من الحروف الى التواريخ المقابلة . ويمكن ملاحظة ذلك من الأمر التالى :

```
@5,5 GET mdate RABGE CTOD('02/11/88),CTOD('08/08/89')
```

وعندما يقوم المستخدم بادخال أى قيمة عددية أو تاريخية خارج المدى الذى تم تحديده . يقوم برنامج (+ DBase III) بعرض المدى المسموح باستخدامه على عمود الحالة (Status bar) أو على لوحة الأهداف (Scoreboard) كما يقوم بارشاد المستخدم الى الضغط على مسطرة المسافات (Space Bar) لمسح القيمة التى سبق ادخالها والمحاولة مرة ثانية .

١٧ - ٥ استخدام التعبير (TRANSFORM)

يستخدم التعبير (PICTURE) مع الأوامر (SAY ... @) ، (GET ... @) فقط ، ولكن لايمكن استخدامه مع باقى أوامر برنامج (+ DBase III) مثل الأوامر :

```
? , ?? , DISPLAY , LABEL , LIST , REPORT
```

لذلك يستخدم التعبير (TRANSFORM) فى تحديد شكل المدخلات مع هذه الأوامر . حيث يتم كتابة التعبير (TRANSFORM) يليه المتغير المراد تحديد شكله . ثم الدالة والرموز المستخدمة فى تحديد الشكل المطلوب . والأمر التالى يوضح هذه العملية :

```
LIST TRANSFORM (mname , "@R X X X X X X X X ")
```

وهذا التعبير (TRANSFORM) يكــون مــفيدا جــدا عند استخدام الأمر

(CREATE REPORT) فى تصميم التقارير المطلوبة للبرنامج .

ملاحظة

ماسبق ذكره فى هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثـــل DBase IV ، FoxBase ، FoxBase+ ، FoxPro .

الباب الثامن عشر

الدوال المستخدمة مع المدخلات

عند عرض البيانات على الشاشة يتطلب الأمر في بعض الأحيان عرض بيانات حقول أو متغيرات ذاكرة مختلفة النوع . فمثلا يمكن عرض بيانات حقل حرفي مع حقل عددي أو تاريخي . وبرنامج (+ DBase III) لايسمح بكتابة أنواع مختلفة من الحقول أو متغيرات الذاكرة ذات الأنواع المختلفة على نفس السطر في البرنامج الذى يتم كتابته . لذلك يلزم اجراء عمليات تحويل للبيانات الموجودة في الحقول أو متغيرات الذاكرة . وتستخدم مجموعة من الدوال في التحويل من نوع الى آخر . كما أن هناك دوال تؤدى وظائف أخرى يتم القاء الضوء عليها في الأجزاء التالية .

١٨ - ١ الدوال الحرفية

١٨ - ١ - ١ استخدام الدالة (STR)

تستخدم هذه الدالة في تحويل أى قيمة عددية الى الحروف المقابلة (String) فمثلا عند كتابة الأمر التالى :

```
? STR(500)
```

والضغط على مفتاح الادخال يتم تحويل العدد (500) الى الحروف المقابلة كالآتي :

```
"  500 "
```

ويلاحظ هنا وجود مسافات خالية قبل العدد . وذلك لأن الدالة تعطى سلسلة حرفية (Character String) طولها عشرة حروف .

ويمكن التحكم في طول السلسلة الحرفية ، وكذلك في عدد الأرقام العشرية عن طريق اضافة عددين بعد العدد المطلوب تحويله . العدد الأول يحدد طول السلسلة (String) ، والعدد الثاني يحدد عدد الأرقام العشرية (Decimal Numbers) . فمثلا يمكن كتابة الأمر التالى :

```
? STR(11.14,5,2)
```

ويلاحظ في هذه الحالة ظهور العدد كالآتي :

```
11.14
```

ويجب ملاحظة أن الأمر (?) هو أمر من أوامر (+ DBase III) ويؤدى الى اظهار نتيجة أى علاقة تلى الأمر . وهو يعنى الاستفهام عن قيمة

علاقة معينة .

ويلاحظ من المثال السابق أن طول سلسلة الحروف تم تحديده عن طريق الرقم (5) ، وذلك أخذا في الاعتبار أن نقطة الكسر العشرى (Decimal Point) تحسب ضمن طول السلسلة . والأمر التالى يوضح استخدام الأمر (STR) بشكل آخر

```
STORE 11.15 To X
STORE STR(X*10,5) To Y
?
```

يلاحظ عند كتابة الأمر الأخير والضغط على مفتاح الادخال ظهور السلسلة الحرفية التالية :

```
111
```

ويلاحظ فى هذه الحالة اختفاء الكسر العشرى ، وذلك لأن الدالة لم تتضمن العدد الذى يحدد عدد الأرقام العشرية .

أما اذا أريد اظهار الكسر العشرى كما هو ، فيتم كتابة الأمر التالى :

```
STR(X*10, 5, 2)
```

يلاحظ فى هذه الحالة ظهور العدد كالآتى

```
111.5
```

وعند وجود أى تناقض فى المعاملات الخاصة بالدالة (STR) ، فان البرنامج يعطى رسالة خطأ (Error message) . فمثلا عند كتابة الأمر التالى :

```
? STR(NUM,1,2)
```

والضغط على مفتاح الادخال تظهر رسالة خطأ . وذلك لأن عدد الأرقام العشرية أكبر من العدد الكلى للأرقام الموجودة فى العدد .

١٨ - ١ - ٢ استخدام الدالة (VAL)

تستخدم الدالة (VAL) فى تحويل البيانات الحرفية (Strings) الى قيم عددية . وهى عكس العملية التى تقوم بها الدالة (STR) . فمثلا عند

كتابة الأوامر التالية :

```
STORE '886.67' TO string
? VAL(string)
```

مع الضغط على مفتاح الادخال بعد كل أمر ، يلاحظ ظهور العدد التالي :

```
886.67
```

ومع أن العدد لم يتغير الا أنه أصبح قيمة عددية يمكن اجراء أى عمليات حسابية عليها .

ملاحظة

عند استخدام الدالة (VAL) مع بيانات حرفية (Strings). لاتحتوى على أعداد فان البرنامج يعطى القيمة صفر . فمثلا عند كتابة الأمر التالى :

```
? VAL('Hello')
```

والضغط على مفتاح الادخال يلاحظ ظهور القيمة صفر .

١٨ - ١ - ٣ مقارنة البيانات الحرفية

رغم أن المقارنة دائما ترتبط فى الذهن بالقيم العددية ، حيث أنها القيم التى يسهل مقارنتها واستنتاج القيمة الأكبر أو القيمة الأصغر، الا أن الحروف والبيانات الحرفية أيضا يمكن مقارنتها . وتستخدم هذه المقارنة فى بعض البرامج التى يتم كتابتها بواسطة برنامج (+ DBase III) أو برامج عائلة (DBase)الأخرى مثل (DBase IV) ، (+ FoxBase) ، (FoxPro) . وتعتمد هذه المقارنة على أن كل حرف له كود الآسكى الخاص به . فعند مقارنة حرفين ، يتم مقارنة العدد الممثل لكود الآسكى (ASCII Code) فى كلا الحرفين . فمثلا عند كتابة الأمر التالى :

```
? 'A' <'a'
```

والضغط على مفتاح الادخال ، تظهر القيمة .T. أى (True). وذلك لأن كود الآسكى الخاص بالحرف (A) وهو (65) أصغر من كود الآسكى الخاص بالحرف (a) وهو (97).

وبالمثل يمكن كتابة الأمر التالى :

```
? '950'>'750'
```

ويجب ملاحظة أن المقارنة هنا بين بيانات حرفية رغم أنها تحتوى على أعداد . فعند الضغط على مفتاح الادخال تظهر نتيجة المقارنة وهى .T. أى (True) . وذلك لأن كود الأسكى الخاص بالرقم ٩ وهو (57) أكبر من كود الأسكى الخاص بالرقم (7) وهو (55) .

ويمكن مقارنة بيانات حرفية باستخدام معامل التساوى (=) ، مع ملاحظة أن المقارنة تتم حرفا حرفا حتى تنتهى السلسلة الحرفية (String) الموجودة يمين علامة التساوى . وفى هذه الحالة يعطى البرنامج القيمة (True) أى صحيح .

فمثلا عند كتابة الأمر التالى :

```
? 'abcd'='abc'
```

والضغط على مفتاح الادخال تظهر القيمة .T. أى (True) .

أما عند كتابة الأمر التالى :

```
? 'abc'='abcd'
```

والضغط على مفتاح الادخال ، فتظهر القيمة .F. أى (Fasle) . وذلك لأن الحروف يسار علامة التساوى تنتهى قبل الحروف يمينها .

تحـــذير

يجب ملاحظة عدم مقارنة قيم مختلفة فى النوع .

فمثلا عند كتابة الأوامر التالية :

```
today = DATE()
IF today = "01/11/88"
```

يتوقف البرنامج . وذلك لأن المتغير (today) يحتوى على قيمة تاريخية ، فى حين القيمة الموجودة يمين علامة التساوى ("01/11/88") هى قيمة حرفية . فاذا أريد علاج هذا الخطأ يتم تحويل نوع أحد

القيمتين الى نسوع القيمة الأخرى كما سيتم الايضاح فى الجزء الخاص بتحويل القيم التاريخية .

١٨ - ١ - ٤ استخدام الدالة (LEN)

فى بعض الأحيان يحتاج مخطط البرامج الى تحديد طول سلسلة حرفية (String). فمثلا عندما يقوم المستخدم بادخال بيانات معينة ، يمكن اختبار طول السلسلة الحرفية (String) التى قام بكتابتها للتأكد من صحة هذه البيانات . وتستخدم لذلك الدالة (()LEN) . وهذه الدالة ينتج عنها قيمة عددية تمثل طول هذه السلسلة . فمثلا اذا كان هناك متغير عددى (String) عدد حروفه ٢٠ حرفا . فعند استخدام الأمر التالى :

```
? LEN(string)
```

والضغط على مفتاح الادخال يظهر على الشاشة الرقم ٢٠ .

وعند كتابة الأمر التالى :

```
? LEN("Mohamed")
```

والضغط على مفتاح الادخال يظهر الرقم (7) .

والأوامر التالية توضح استخدام هذا الأمر داخل برنامج .

```
IF LEN(Input)<5
    Do Error & & Branch to Error.prg
ELSE
   ------
   ------
   ------     commands
   ------
   ------
ENDIF
```

وفى هذه الأوامر يقوم البرنامج باختبار طول السلسلة الحرفية الموجودة فى المتغير (Input) فاذا كان أقل من (5) يتم التفرع الى البرنامج (Error)

١٨ - ١ - ٥ استخدام الدالة (SUBSTR)

تستخدم هذه الدالة للحصول على جزء من سلسلة الحروف (String). ويجب فى هذه الحالة ابلاغ البرنامج عن مكان البداية وطول السلسلة المطلوب الحصول عليها . فمثلا اذا كان هناك متغير اسمه (String1) يحتوى على الحروف التالية :

```
"My name is HASAN"
```

فعند كتابة الأمر التالى والضغط على مفتاح الادخال :

```
? SUBSTR(string1,1,10)
```

يلاحظ ظهور الآتى :

```
My name is
```

وذلك لأن العدد الأول (1) يحدد بداية السلسلة المطلوبة والعدد الثانى (10) يحدد عدد حروف هذه السلسلة . ويجب ملاحظة أن المسافات الخالية أيضا تؤخذ فى الاعتبار .

وعند كتابة عدد واحد بعد الدالة فان ذلك يعنى أن السلسلة المطلوبة تبدأ من هذا العدد وتنتهي بنهاية السلسلة الأصلية . فمثلا عند كتابة الأمر التالى والضغط على مفتاح الادخال :

```
? SUBSTR(string1,12)
```

يلاحظ ظهور الآتى :

```
HASAN
```

ويمكن استخدام هذه الدالة فى اعادة استخدام سلسلة من الحروف فى عدة أماكن من البرنامج . وذلك عن طريق تخزين هذه السلسلة فى متغير ذاكرة واستخدام أى جزء منها حسب الحاجة . فمثلا فى المثال السابق عندما يــراد استخدام جــزء من السلسلة مع اسم آخر يستخدم الأمر التالى :

```
? SUBSTR(string1,1,10) +"MOHAMED"
```

فعند الضغط على مفتاح الادخال ، يلاحظ ظهور الآتى :

```
My name is MOHAMED
```

ويلاحظ في هذه الحالة ظهورالاسم (MOHAMED) مكان الاسم (HASAN) .

١٨ - ١ - ٦ الدالة (LEFT) والدالة (RIGHT)

تستخدم الداله ()LEFT للحصول على جزء من سلسلة حرفية بدءا من يسار السلسلة وبعدد معين من الحروف . فمثلا في المثال السابق يمكن استخدام الدالة كالآتي :

```
? LEFT(string1,10)
```

وعند الضغط على مفتاح الادخال ، يظهر الآتي :

```
My name is
```

أما الداله ()RIGHT فتؤدى الى الحصول على جزء من سلسلة حرفية بدءا من يمين السلسلة وبعدد الحروف المحدد بالرقم الموجود مع الدالة . فمثلا في المثال السابق يمكن استخدام الدالة كالآتي :

```
? RIGHT(string1,5)
```

وعند الضغط على مفتاح الادخال يلاحظ ظهور الآتي :

```
HASAN
```

١٨ - ١ - ٧ استخدام الدالة (AT)

تستخدم هذه الدالة في تحديد مكان مجموعة من الحروف داخل سلسلة حرفية . فمثلا في المثال السابق عندما يراد تحديد مكان كلمة (HASAN) داخل سلسلة حرفية يمكن كتابة السطر التالي :

```
? AT('HASAN',string1)
```

وعند الضغط على مفتاح الادخال يظهر الرقم (12) وهو يعني أن الاسم يبدأ من الحرف رقم ١٢ .

ويلاحظ أن الحروف المطلوب تحديد مكانها توضع بين علامات تنصيص (Quotation).

١٨ -١- ٨ استخدام الدالة (UPPER) والدالة (LOWER)

تستخدم الداله ()UPPER في تحويل سلسلة حرفية من حروف صغيرة (Lowercase) الى حروف كبيرة (Uppercase) . كما تستخدم الدالة ()LOWER في عمل العكس ، أي تحويل الحروف الكبيرة الى حروف صغيرة . فمثلا عند استخدام الداله ()UPPER مع المتغير الحرفي (string1) كالآتي:

```
? UPPER(string1)
```

يلاحظ ظهور الآتي :

```
MY NAME IS HASSAN
```

كما يمكن جعل أول حرف كبيرا (Uppercase) وباقي الحروف صغيرة . وذلك بكتابة السطر التالي :

```
'M' + LOWER(SUBSTR(string1,2)
```

وعند الضغط على مفتاح الادخال يلاحظ ظهور الآتي :

```
My name is hasan
```

ويلاحظ من هذا المثال أنه تم اضافة الحرف (M) الى جزء من السلسلة الحرفية الموجودة في المتغير (string1) يبدأ من الحرف رقم (2) وهو (Y) وحتى آخر السلسلة الحرفية . كما تم تحويل هذا الجزء الى حروف صغيرة (Lowercase) باستخدام الدالة (LOWER).

والمثال التالي يوضح استخدام الدالة (UPPER) داخل برنامج :

```
@ 15,15 SAY "Send report to printer Y/N" ;
GET Answer
IF UPPER(Answer) = "Y"
   SET PRINT ON
ENDIF
```

وفي هذا المثال يتم سؤال المستخدم اذا كان يريد طباعة التقرير أم لا . فاذا كان يريد الطباعة فانه يكتب (Y) . وفي هذه الحالة اذا كتب المستخدم حرف y صغيرا أو كبيرا لايؤثر ذلك على البرنامج . وذلك لأن

الدالة (UPPER) تقوم بتحويل هذا الحرف الى حرف كبير (Uppercase) .

١٨-١-٩ استخدام الدوال (TRIM) ، (LTRIM) ، (RTRIM)

عند ادخال مدخلات حرفية في بعض الحقول أو متغيرات الذاكرة ، ففي معظم الأحيان يكون طول السلسلة الحرفية التي يتم ادخالها أصغر من طول الحقل أو متغير الذاكرة . وفى هذه الحالة تكون هناك مسافات خالية (Spaces) بعد السلسلة الحرفية . فمثلا قد يتم ادخال اسم مثل ("Tarek Mohmoud") في حقل أو متغير ذاكرة طوله ٢٠ حرفا . في هذه الحالة تبقى مسافات خالية بعد الاسم . وفى معظم الأحيان يريد مخطط البرامج التخلص من هذه المسافات الخالية والحصول على الحروف فقط . وذلك عندما يريد كتابة تقرير مثلا وضم مجموعة من الحقول فى سطر واحد أو عندما يتم انشاء حقل للاسم الأول وحقل للاسم الثاني ، فى هذه الحالة يجب التخلص من المسافات الخالية من حقل الاسم الأول حتى يظهر الاسم بصورة مقبولة . وفى هذه الأحوال وغيرها يتم استخدام الدالة ()TRIM . هذه الدالة تؤدى الى التخلص من المسافات الخالية بعد نهاية السلسلة الحرفية . ولتوضيح تأثير هذه الدالة يتم دراسة المثال الآتى :

نفرض أنه تم انشاء حقل للاسم الأول اسمه (Fname) وحقل للاسم الثاني اسمه(Sname) . ونفرض أن الحقل (Fname) طوله عشرة حروف ، والحقل (Sname) طوله عشرة حروف . فعند عرض اسم مثل (Aly Hasan) مثلا يتم كتابة السطر التالى :

```
? Fname + Sname
```

وعند الضغط على مفتاح الادخال يلاحظ ظهور الاسم كالآتى :

```
Aly         Hasan
```

ويحدث هذا لأن الحقل (Fname) طوله عشرة حروف ، وتم كتابة ٣ حروف فقط منه . أى أن هناك سبعة مسافات خالية (Sapces) . فى حين يمكن استخدام الدالة (TRIM) كالآتى :

```
? TRIM(Fname)+Sname
```

وعند الضغط على مفتاح الادخال يلاحظ ظهور الآتى :

AlyHasan

ويلاحظ عدم وجود أي مسافة بين الاسمين .. فاذا أريد كتابة الاسمين وبينهما مسافة يتم كتابة السطر التالي :

```
? TRIM(Fname)+" "+Sname
```

وعند الضغط على مفتاح الادخال ، يلاحظ ظهور الآتي :

```
Aly Hasan
```

أما الداله LTRIM() فتستخدم في التخلص من المسافات الخالية في أول السلسلة الحرفية . ويفيد هذا في عدة حالات منها على سبيل المثال التحقق من صحة ادخال المستخدم للمدخلات الحرفية ، وعدم ادخال مسافة خالية في بداية أى سلسلة حرفية . حيث أن كتابة مسافة خالية في أول أى حقل حرفي يمكن أن تسبب مشاكل كثيرة في الاسترجاع أو البحث عن سجل معين .

فمثلا فى المثال السابق ، نفرض أن المستخدم عند كتابته الاسم (ALY) قام بالضغط على مسطرة المسافات قبل الاسم عن طريق الخطأ . فى هذه الحالة يظهر الاسم كالآتى مثلا :

```
-ALY - - - - - - -
```

في هذه الحالة يتم تخزين الاسم وفى أوله مسافة خالية (Space) . وعندما يراد التخلص من هذه المسافة يستخدم الأمر (LTRIM) كالآتى :

```
? LTRIM(Fname)
```

وعند الضغط على مفتاح الادخال يظهر الاسم كالآتى :

```
ALY - - - -
```

أما الدالة (RTRIM) فانها تؤدى نفس عمل الدالة (TRIM) ، لأنها تؤدى الى التخلص من المسافات الموجودة يمين سلسلة الحروف .

ملاحظة

عندما يراد التأكد من التخزين الصحيح للبيانات التى يكتبها المستخدم ،

والتغلب على الأخطاء التى قد تنتج عن كتابة مسافات خالية فى أول أو آخر الاسم . يمكن استخدام متغير الذاكرة أولا فى استقبال مدخلات المستخدم . ثم يتم استخدام الدالة (TRIM) والدالة (LTRIM) مع هذا المتغير . وذلك قبل نقله فعلا الى مكانه فى الملف .

فمثلا اذا كان هناك حقل للاسم (name) يتم انشاء متغير ذاكرة اسمه (mname) . وذلك كالآتى :

```
STORE SPACE(30) TO mname
```

ثم يتم كتابة السطر التالى للتخلص من المسافات فى أول الاسم وآخره كالآتى :

```
STORE LTRIM(TRIM(mname)) TO mname
```

فى هـــذه الحالة يتم تخزين مايدخله المستخدم فى متغير الذاكرة (mname) بدون أى مسافات فى أوله أو آخره .

وتستخدم الدالة (TRIM) أيضا فى الحصول على الطول الصحيح لأى سلسلة حرفية باستخدام الدالة (LEN) . حيث أن مجرد استخدام الدالة (LEN) يمكن أن يعطى نتيجة خاطئة نتيجة وجود مسافات خالية (Spaces) فى أول السلسلة الحرفية أو فى آخرها . ولكن مثلا عند كتابة الآتى :

```
LEN (LTRIM(TRIM(mname)))
```

فى هذه الحالة يتم الحصول على الطول الحقيقى لهذا الاسم بدون أى مسافات فى أوله أو فى آخره .

١٨-١-١٠ جمع البيانات الحرفية (CONCATINATION)

يمكن جمع بيانات حرفية (String) على بيانات حرفية أخرى باستخدام علامة الجمع (+) . ويؤدى هذا الى تكوين سلسلة حرفية جديدة (String) تحتوى على السلسلة الحرفية الأولى يليها السلسلة الحرفية الثانية . فمثلا عندما يكون هناك حقل حرفى اسمه (Account_No) ، ويراد جمع بيانات الحقل على سلسلة حرفية أخرى (String) يمكن استخدام الأمر التالى :

```
? "The account number is"+ Account_no
```

فاذا افترضنا أن الحقل (Account_No)يحتوى على العدد الآتى (5788) . فان تنفيذ الأمر السابق يؤدى الى ظهور الآتى :

```
The account number is5788
```

ويلاحظ في هذه الحالة التصاق السلسلة الحرفية الأولى بالسلسلة الحرفية الثانية بدون أى مسافات .

ويمكن التغلب على ذلك بترك مسافة خالية فى آخر السلسلة الحرفية بعد كلمة (is) كالآتى مثلا :

```
? "The account no is  " + Account_no
```

ويلاحظ في هذه الحالة اضافة مسافة خالية بعد السلسلة الأولى . كما يمكن تنفيذ ذلك بطريقة أخرى بكتابة الأمر التالى :

```
? "The account no is" + " " + Account_no
```

ويلاحظ في هذه الحالة جمع سلسلة حرفية أخرى تحتوى على مسافة خالية بين السلسلة الأولى والسلسلة الثانية .

وهناك طريقة أخرى لجمع السلاسل الحرفية (strings) وذلك باستخدام علامة (-). والجمع بواسطة علامة (-) يؤدى الى نقل المسافات الخالية الخلفية (Trailing blanks) من السلسلة الأولى الى نهاية السلسلة الثانية . فمثلا فى المثال الخاص بالاسم عند جمع الحقل الخاص بالاسم الأول (Fname) الى الحقل الخاص بالاسم الثانى (Sname) باستخدام علامة (-) يلاحظ ضم الاسمين وانتقال المسافات الخالية من الاسم الأول الى نهاية الاسم الثانى .

١٨-١-١١ التحويل بين الحروف وكود الآسكى

يتيح برنامج (+ DBase III) لخطط البرامج الحصول على أى حرف عن طريق كود الآسكي (ASCII Code) الخاص بهذا الحرف ، وذلك باستخدام الدالة (CHR) . ويساعد هذا فى حالات كثيرة منها مثلا التحكم فى شكل شاشة الادخال . حيث يمكن الحصول على بعض الحروف الخاصة التي تساعد على رسم الخطوط والاشكال التي يمكن من خلالها رسم شاشة الادخال . فمثلا الدالة (CHR(205 تساعد في رسم مستطيل على الشاشة . وكذلك الدالة (CHR(201 تساعد فى رسم أركان هذا المستطيل

وهكذا .

كما يمكن الحصول على أى حرف عن طريق كود الآسكى الخاص به فمثلا للحصول على حرف (a) يتم كتابة السطر التالى :

```
? CHR(97)
```

وعند الضغط على مفتاح الادخال يظهر الحرف (a) .

كذلك يمكن الحصول على كود الآسكى (ASCII Code) الخاص بأى حرف باستخدام الدالة (ASC) . فمثلا للحصول على كود الآسكى الخاص بالحرف (a) يتم كتابة السطر التالى :

```
? ASC('a')
```

وعند الضغط على مفتاح الادخال يظهر العدد (97). كما يمكن الحصول على أى حرف عن طريق حرف آخر كالآتى مثلا :

```
? CHR(ASC('a')+1)
```

وعند الضغط على مفتاح الادخال يظهر الحرف (b) وذلك لأن كود الآسكى الخاص بالحرف (b) يزيد عن كود الآسكى الخاص بالحرف (a) بمقدار (1). وكذلك بالنسبة لجميع الحروف الهجائية . حيث أن كل حرف يزيد كود الآسكى الخاص به عن الحرف الذى يسبقه بمقدار (1).

كما أن هناك بعض أرقام الآسكى (ASCII Code) التى تؤدى الى حدوث تأثيرات معينة مثل تشغيل الجرس (Bell) .

فمثلا عند كتابة الأمر التالى والضغط على مفتاح الادخال

```
? CHR(7)
```

يلاحظ تشغيل الجرس (Bell) .

ويمكن عن طريق ذلك توجيه انتباه المستخدم عند حدوث خطأ مثلا أو فى أى حالات أخرى مشابهة .

١٨ - ٢ الدوال العددية

هناك عدة دوال عددية يتيح برنامج (+ DBase III) لمخطط البرامج استخدامها داخل البرنامج ويتم القاء الضوء عليها فى الأجزاء التالية .

كما يمكن الرجوع الى الملحق الموجود في آخر الكتاب لمعرفة الدوال الخاصة ببرامج عائلة (DBase) الأخرى مثل (DBase IV) ، (+ FoxBase) ، (FoxPro) .

١٨ - ٢ - ١ الدالة (ABS)

وتستخدم هذه الدالة فى الحصول على القيمة المطلقة لأى عدد . ويمكن استخدامها فى تحديد الفرق العددي بين قيمتين عدديتين دون الحاجة لمعرفة أيهما أكبر من الأخرى . فمثلا يمكن ملاحظة الأوامر التالية :

```
i = 20
j = 80
? ABS(i-j)
```

وعند الضغط على مفتاح الادخال يلاحظ ظهور العدد ٦٠ .

١٨ - ٢ - ٢ الدالة (EXP)

وتستخدم هذه الدالة فى الحصول على القيمة الأسية (e^x) .فمثلا للحصول على النسبة التقريبية (e) يتم كتابة الأمر التالى :

```
? EXP(1)
```

والضغط على مفتاح الادخال فيلاحظ ظهور العدد الآتى :

```
(2.718)
```

١٨ - ٢ - ٣ الدالة (INT)

وتستخدم هذه الدالة فى الحصول على العدد الصحيح فى قيمة عددية معينة . فمثلا عند تخزين العدد (15.52) فى المتغير (x) ثم كتابة الأمر التالى والضغط على مفتاح الادخال :

```
? INT(X)
```

يلاحظ ظهور العدد (15)

١٨ - ٢ - ٤ الدالة (LOG)

وتستخدم هذه الدالة في الحصول على اللوغاريتم الطبيعي لأى عدد . وهي نادرا ما تستخدم في برامج قواعد البيانات . ولكن قد تكون هناك بعض التطبيقات الرياضية التى تتطلب استخدامها .

١٨ - ٢ - ٥ الدالة (MAX)

وتستخدم هذه الدالة في الحصول على أكبر قيمة من قيمتين . ويمكن استخدامها فى تحديد القيم التى تزيد عن قيمة معينة . وذلك عن طريق تحديد قيمة معينة يراد اعتبارها الحد الأدنى للقيم الموجودة فى حقل معين. ثم مقارنة جميع القيم الموجودة فى الحقل بالقيمة التى تم تحديدها، للحصول على جميع القيم التى تزيد عن هذه القيمة .

١٨ - ٢ - ٦ الدالة (MIN)

وتستخدم هذه الدالة في الحصول على أصغر قيمة من قيمتين . ويمكن استخدامها في تحديد الحد الأعلى للقيم الموجودة فى حقل معين وتحديد القيم التى تقل عن هذا الحد .

١٨ - ٢ - ٧ الدالة (MOD)

وتستخدم هذه الدالة لتحديد المقدار الباقي بعد قسمة عدد على عدد آخر . وتستخدم في التحويل من نوع من الوحدات الى نوع آخر . فمثلا لتحويل عدد من الدقائق يساوي ٣٦٥٠٠ الى مايقابله من أيام وساعات ودقائق يمكن استخدام الأوامر التالية :

```
t = 36500
minutes = MOD(t,60)
h = INT(t/60)
hours = MOD(h,24)
days = INT(h/24)
? t,"minutes are",days,"days",hours,"hours";
minutes,"minutes"
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
36500 minutes are:25 days 8 hours 20 minutes
```

وذلك لأن السطر الثاني MOD(t,60) يؤدى الى ظهور باقى قسمة العدد (36500) على (60) .

والسطر الثالث INT(t/60) يؤدى الى ظهور العدد الصحيح الناتج عن قسمة العــدد (36500) على 60 أى عدد الساعات الكلية .

والسطر الرابع MOD(h,24) يؤدى الى حساب العدد الباقى من قسمة عدد الساعات الكلية عـلى (24) . أى عدد الساعات المتبقية من الأيام .

والسطر الخامس INT(h/24) يؤدى الى ظهور عدد الأيام الصحيحة .

والسطر السادس يؤدى الى عرض الأعداد الممثلة للأيام والساعات والدقائق .

١٨ - ٢ - ٨ الدالة (ROUND)

وتستخدم هذه الدالة لعمل تقريب للكسر العشرى بعد تحديد عدد معين من الكسور العشرية المطلوب ظهورها فى العدد . فمثلا عند كتابة الأمر التالى :

```
?     ROUND(15.847321,2)
```

والضغط على مفتاح الادخال يظهر الآتى :

```
15.85
```

١٨ - ٢ - ٩ الدالة (SQRT)

وتستخدم هذه الدالة للحصول على الجذر التربيعى لأى عدد .

١٨ - ٣ الدوال التاريخية (Date Functions)

يتعامل البرنامج مع التواريخ عن طريق تمثيل كل تاريخ بعدد معين . ورغم أن

البرنامج يعرض التاريخ على الشاشة على الشكل المعروف بالنظام الأمريكى (MM/DD/YY) ، والذى يعنى رقمين للشهر (MM) ورقمين لليوم (DD) ورقمين للسنة (YY) ، الا أن البرنامج يتعامل مع العدد الممثل لهذا التاريخ فقط . وعن طريق ذلك العدد يمكن طرح تاريخ من تاريخ للحصول على عدد الأيام المحصورة بين التاريخين . كما يمكن اضافة عدد من الأيام الى تايخ معين للحصول على تاريخ جديد أو طرح عدد من الأيام من تاريخ معين للحصول على تاريخ جديد . وهكذا .

كما يوفر البرنامج عددا من الدوال التى تستخدم فى التعامل مع التواريخ مثل الدالة (DATE()) . هذه الدالة تعطى دائما تاريخ اليوم الحالى الذى يتم ادخاله عند بدء تشغيل نظام التشغيل (MS-DOS) .

فمثلا عندما يكون هناك ملف حسابات (Accounts) ، وهناك مجموعة من العملاء الذين يحل ميعاد دفع الدين الخاص بهم فى يوم محدد وليكن هذا اليوم هو (01/01/1990) وهو اليوم الحالى . فيمكن كتابة السطر التالى :

```
LIST FOR Date = DATE()
```

فى هذه الحالة تظهر قائمة بأسماء الأشخاص الذين يحل موعد سدادهم الدين فى هذا اليوم .

كما يمكن تحديد موعد سداد قرض بعد خمسين يوما من اليوم الحالى كالآتى مثلا :

```
STORE DATE() + 50 TO overdue
```

حيث يتم انشاء متغير ذاكرة (Overdue) يتم تخزين موعد سداد القرض فيه . ثم اختبار تاريخ السداد بعد خمسين يوما عن طريق السطور التالية من البرنامج :

```
IF overdue = DATE()
Do letter
ENDIF
```

فى هذه الحالة يتم تشغيل برنامج (Letter) عندما يكون تاريخ اليوم الحالى (Date()) مساويا لتاريخ الدفع (Overdue) .

وهناك دوال تساعد على تحديد ترتيب اليوم فى الأسبوع أو فى الشهر أو تحديد ترتيب الشهر فى السنة ، أو تحديد السنة نفسها . فمثلا بالنسبة للتاريخ (01/28/1990) يمكن الحصول على ترتيب اليوم فى الأسبوع كالآتى :

```
? DOW(DATE())
```

حيث تعطي الدالة (DOW) اليوم المقابل للتاريخ . فعند الضغط على مفتاح الادخال يلاحظ ظهور العدد (1) الذى يدل على أن اليوم هو الأحد ، حيث أن يوم الأحد يمثل أول أيام الأسبوع .

كما يمكن الحصول على ترتيب اليوم فى الشهر كالآتى :

```
? DAY(DATE())
```

وعند الضغط على مفتاح الادخال يظهر العدد (28) الذى يدل على أن اليوم هو الثامن والعشرون من الشهر .

كما يمكن الحصول على ترتيب الشهر فى السنة كالآتى :

```
? MONTH(DATE())
```

وعند الضغط على مفتاح الادخال يظهر العدد (1) الذى يدل على أن الشهر هو شهر يناير .

كما يمكن الحصول على السنة كالآتى :

```
? YEAR(DATE())
```

وعند الضغط على مفتاح الادخال يظهر العدد (1990) .

١٨-٣-١ تحويل التاريخ الى حروف (DATE TO CHARACTER)

كما سبق الايضاح فان التواريخ يتم تمثيلها فى البرنامج كأعداد . فعندما يراد استخدام هذه التواريخ فى سلاسل حرفية (Strings) . فى هذه الحالة يلزم أولا تحويل هذه التواريخ الى حروف (Characters) . وتستخدم لذلك الدالة DTOC() .

فمثلا عند كتابة السطر التالى :

```
? DTOC(DATE())
```

و الضغط على مفتاح الادخال ، يلاحظ ظهور التاريخ الحالى كالآتى :

```
01/28/90
```

تحـــذير

لا يـــمكن اجراء حسابات على التاريخ وهو فى صورة سلسلة حرفية (String) ، مثل اضافة أو طرح عدد من الأيام من التاريخ أو طرح تاريخ معين من هذا التاريخ . ولكن يلزم أولا تحويله مرة ثانية الى تاريخ غير حرفى ، كما سيتم الايضاح فيما بعد .

وهناك دوال أخرى يمكن استخدامها فى عرض أيام الأسبوع بالحروف بدلا من الأرقام كالآتى مثلا :

```
? CDOW (DATE())
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
Sunday
```

كما يمكن عرض الشهور بالحروف أيضا كالآتى :

```
? CMONTH(DATE())
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
January
```

وتستخدم هذه الدوال فى عرض التواريخ بأى صورة مطلوبة من خلال البرنامج . فمثلا يمكن كتابة السطر التالى :

```
?  CDOW(DATE())  +',' + CMONTH(DATE())  +' '+;
LTRIM(STR(DAY(DATE()),2))+','+STR(YEAR(DATE()),4)
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
Sunday , January 12,1990
```

وعند ادخال هذا الأمر فى البرنامج فانه ينفذه على تاريخ اليوم الحالى حسب تاريخ اليوم الذى يتم فيه تشغيل البرنامج .

١٨-٣-٢ تحويل الحروف الى تاريخ (CHARACTER TO DATE)

عندما يراد استخدام التواريخ الحرفية كتواريخ مفهومة بالنسبة لبرنامج (+ DBase III) ، وذلك حتى يمكن استخدام هذه التواريخ في العمليات الحسابية المختلفة كما سبق الايضاح . في هذه الحالة تستخدم الدالة ()CTOD التى تقوم بتحويل التاريخ الحرفى الى التاريخ المقابل . فمثلا عند كتابة الأوامر التالية :

```
STORE '1/20/90' TO string
STORE CTOD(string) TO newday
```

فالسطر الأول يؤدي الى تكوين متغير ذاكرة اسمه (string) يحتوى على السلسلة الحرفية (1/20/90) .

أما السطر الثاني فانه يؤدي الى تكوين متغير تاريخي نتيجة تحويل المتغير الحرفى الى متغير تاريخى .

ويجب ملاحظة أن المتغير (newday) في هذه الحالة يحتوى على نفس الأرقام (1/20/90) ولكنها تمثل شيئا مختلفا عن الأرقام الموجودة في متغير الذاكرة الحرفى (string) .

وعندما يراد انشاء متغير تاريخى غير تاريخ اليوم يمكن استخدامه داخل البرنامج ، يستخدم السطر التالى :

```
Date = CTOD('01/10/90')
```

ويمكن بعد ذلك ادخال أي تاريخ فى هذا التغير . كما يمكن انشاء متغير تاريخى خال (blank) كالآتى :

```
Date1 = CTOD(' / / ')
```

١٨ - ٣ - ٢ استخدام التواريخ فى المقارنة (Comparison)

عند عمل مقارنة بين تاريخين فان هذين التاريخين يجب الا يكونا حرفيين . أى يلزم أولا تحويلهما الى تاريخ ثم عمل المقارنة المطلوبة . فمثلا عند كتابة السطر التالى :

```
? '01/01/90'> '12/31/89'
```

وعند الضغط على مفتاح الادخال فان ذلك يعطى (.F.) أى غير صحيح مع أن التاريخ (12/31/89) يسبق التاريخ (01/01/90) أى أن النتيجة يجب أن تكون صحيحة (True) . لذلك يجب أولا تحويل التاريخ الحرفى قبل تنفيذ عملية المقارنة كالآتى :

```
? CTOD('01/01/90') > CTOD('12/31/89')
```

وعند الضغط على مفتاح الادخال تظهر النتيجة (.T.) أى صحيح (True).

وعندما يراد مثلا عرض بيانات الأشخاص الذين يحل موعد سدادهم الدين فى التاريخ (01/28/90) يتم كتابة السطر التالى :

```
DISPLAY ALL FOR overdue = CTOD('01/28/90')
```

١٨ - ٣ - ٤ استخدام الدالة (TIME())

يستخدم البرنامج الداله TIME() لاعطاء الوقت الحالى على هيئة (hours:minutes:seconds) ، وهو الوقت الذى يتم ادخاله عند بدء تشغيل الجهاز . فمثلا عند كتابة الأمر التالى :

```
? TIME()
```

والضغط على مفتاح الادخال يظهر الوقت الآتى مثلا :

```
10:30:35
```

ولتخزين الوقت الحالى داخل حقل قاعدة البيانات ، يتم تعريف هذا الحقل بطول (٨) حروف . ثم يتم استبدال مكونات هذا الحقل بالوقت الحالى . فمثلا اذا كان هناك حقل فى ملف قاعدة البيانات اسمه(Now) . يمكن كتابة السطر التالى :

```
REPLACE Now WITH TIME()
```

فى هذه الحالة يتم ادخال الوقت الحالى فى الحقل Now ، وليكن كالآتى مثلا :

```
10:30:35
```

ويمكن عرض جزء من الوقت مثلا يتضمن الساعات والدقائق فقط ، ولتنفيذ ذلك يتم كتابة السطر التالى :

```
REPLACE Now WITH SUBSTR(Now,1,5)
```

وفى هذه الحالة يحتوى المتغير (Now) على الوقت الآتى :

```
10:30
```

ملاحظة

ماسبق ذكره فى هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، FoxBase + ، FoxPro .

الباب التاسع عشر

مزيد من التحكم فى شاشة الادخال

كما سبق الايضاح فى الباب الخاص بالتحكم في شاشة الادخـال . فان بـرنامـج (+ DBase III) يتيح لمخطط البرامج التحكم في الرسائل التى تظهر على الشاشة وشكل الحقول التى تظهر علـى الشـاشـة باستخدام الأمـر (@...GET) والأمر (@...SAY) ... وهكذا . وفى هذا الباب يتم عرض وسائل أخرى للتحكـم فى الشاشة تتيح لمخطط البرامج تصميم شاشات أحسن وأكثر وضوحا .

١٩ - ١ التحكم فى شكل العمود الضوئى (Highlight)

الوضع المبدئي للأوامر (APPEND, EDIT, GET) هو ظهور أعمدة ضوئية تمثل الحقول المطلوبة . ويحدد كل عمود ضوئى طول الحقل الذى يمثله . واذا أراد المستخدم استخدام علامات تحديد (Delimiters) للبيانات الموجودة فى الحقل ، فانه يستخدم الأمر (SET DELIMITERS ON) . حيث أن الوضع المبدئى لهذه العلامات (DELIMITERS) يكون (OFF).

وعند استخدام الأمر (SET DELIMITERS ON) تظهر علامات (Colons) حول بيانات الحقول . ويمكن استخدام أى علامات أخرى بدلا من علامات (Colons) ، وذلك باستخدام الأمر (SET DELIMITERS TO) ثم كتابة العلامات المطلوب اظهارها بين علامات تنصيص (Quotation) . ويجب ملاحظة أن استخدام علامات التحديد (Delimiters) لاتمنع من ظهور العمود الضوئى . وذلك كالآتى مثلا:



واذا أراد مخطط البرامج الغاء العمود الضوئى ، فانه يستخدم الأمر (SET INTENSITY OFF) .

١٩ - ٢ استخدام العناوين النسبية

كما سبق الايضاح فان استخدام الأمر (@...SAY) والأمر (@...GET) يتيح لمخطط البرامج عرض البيانات أو الرسائل فى أى مكان على الشاشة حسب الاحداثيات التي يتم كتابتها بعد الحرف (@) . ولكـن فـى بعض الأحيان لايكون المكان المطلوب الكتابة فيه محددا . فمثلا عندما يراد عرض بيانات بعض السجلات ، ثم عرض رسالة للمستخدم بعد آخر سطر في البيانات بسطرين مثلا . فى هذه الحالة فان مخطط البرامج لايعرف مقدما عدد السجلات التي سيتم عرض بياناتها . وبالتالى لايعرف المكان الذى يجب عرض الرسالة فيه . لذلك تستخدم الداله ()ROW والدالة ()COL لتحديد العناوين النسبية . وهذا يعنى أن احداثيات نقطة معينة على الشاشة تعتمد على احداثيات آخر نقطة تم الوصول اليها .

والداله ()ROW مثلا تعطى رقم السطر الذى يقف عنده المؤشر فى هذه اللحظة .

فمثلا عند كتابة السطر التالى :

```
@ 5,3 SAY 'Hello Mohamed'
```

فى هذه الحالة فان السطر الحالى (Current row) هو السطر رقم (5) ، كما أن العمود الحالى (Current column) هو العمود رقم ٣ . فعند كتابة الأمر التالى :

```
@ ROW()+2,3 SAY 'How are you?'
```

ثم تنفيذ هذين الأمرين ، يظهر الآتى على الشاشة :

```
Hello Mohamed

How are you?
```

ويلاحظ فى هذه الحالة ظهور الرسالة الثانية بعد سطرين من الرسالة الأولى . وكذلك عند كتابة السطر التالى :

```
@ ROW()+3,COL()+2 SAY 'Well, thank you'
```

وعند تنفيذ الأوامر الثلاثة معا يظهر الآتى على الشاشة :

```
Hello Mohamed

How are you?


                Well, thank you
```

وفى هذه الحالة يلاحظ ظهور الرسالة الثالثة بعد ثلاثة سطور من الرسالة الثانية وبعد عمودين من آخر حرف فى الرسالة الثانية .

ملاحظة

لايمكن استخدام الداله ()ROW والداله ()COL بعد الأمر READ مباشرة . لأن الأمر (READ) يضع المؤشر على السطر رقم ٢٢ والعمود رقم صفر .

ويمكن استخدام العناوين النسبية أيضا في عرض علامات معينة بعد ما يكتبه المستخدم . فمثلا عندما يراد عرض علامة (*) بعد الرسالة (Message) التى يكتبها المستخدم ، يستخدم الأمر التالى :

```
@ 6,24 + LEN(message)  SAY "*"
```

كما يمكن استخدام متغيرات الذاكرة (Memory Variables) فى التحكم فى العناوين النسبية ، وذلك كالآتى مثلا :

```
Line = 4
DO WHILE .NOT. EOF()
     @ Line,10 SAY Name
     @ Line,40 SAY Address
     Line = Line + 1
     SKIP
ENDDO
```

فى هذه الحالة يتم عرض بيانات السجل الأول على السطر الرابع ثم عرض بيانات السجل الثانى على السطر الخامس وهكذا .

١٩-٢ ضبط الحروف فى المنتصف (Centering A String)

عندما يراد كتابة السلسلة الحرفية ابتداء من منتصف السطر فيمكن ببساطة استخدام الأمر (@...SAY) مع كتابة احداثيى نقطة المنتصف بعد الحرف @ .

ولكن عندما يراد وضع ما يكتبه المستخدم فى منتصف السطر فان هذه الطريقة لاتصلح . وذلك لأن مخطط البرامج لايعرف مقدما طول السلسلة الحرفية التى يكتبها المستخدم . وفى هذه الحالة يتم أولا التخلص من المسافات الخالية فى السلسلة الحرفية ثم قسمة طولها على (٢) ثم طرح الناتج من (٤٠) . حيث أن العمود رقم ٤٠ يمثل منتصف الشاشة تماما . وبذلك يتم تحديد النقطة التى يبدأ منها كتابة السلسلة حتى تصبح فى منتصف السطر تماما .

فمثلا اذا كان هناك متغير ذاكرة (mname) ويراد عرض هذا المتغير فى منتصف السطر يتم كتابة السطور التالية :

```
mname = TRIM(mname)
mcenter = 40 - LEN(mname)/2
```

وفى هذه الحالة تم تخزين النقطة التى يجب بدء كتابة الاسم عندها حتى يصبح

فى منتصف السطر تماما فى متغير اسمه (mcenter) . فلكى تتم كتابة هذا الاسم فى منتصف السطر يمكن كتابة الأمر التالى :

```
@ 10,mcenter SAY mname
```

فى هذه الحالة يظهر الاسم فى منتصف السطر بصرف النظر عن طول السلسلة الحرفية .

١٩ - ٤ ضبط الحروف من اليمين (Right Justifying)

عندما يراد كتابة مجموعة من الرسائل (Messages) بحيث تنتهى كلها عند نقطة ثابتة فى اليمين ، يتم انشاء متغير ذاكرة (اسمه width مثلا) ويتم تخزين رقم العمود الثابت (60) مثلا به . ويتم استخدام الأوامر التالية :

```
STORE 60 TO width
@ 5, width-LEN(message1) SAY message1
@ 6, width-LEN(message2) SAY message2
```

فعندما يكون المتغير (message1) محتويا على الرسالة التالية :

```
Enter your name
```

والمتغير (message2) يحتوى على الرسالة التالية :

```
Enter your address
```

فعند تنفيذ الأوامر السابقة يظهر الآتى على الشاشة :

```
   Enter your name
Enter your address
```

١٩ - ٥ حشر حروف داخل السلسلة الحرفية (Stuffing)

عندما يراد حشر مجموعة من الحروف داخل سلسلة حرفية معينة ، فان ذلك يتم باستخدام الدالة ()STUFF .

فمثلا عندما تكون هناك سلسلة حرفية (message1) تحتوى على الرسالة التالية :

" Type Q to quit"

ويراد حشر مجموعة من الكلمات داخل هذه الرسالة حتى تصبح :

" Type R to return to main menu or Q to quit"

فى هذه الحالة يتم أولا تخزين السلسلة الحرفية المطلوب اضافتها فى متغير ذاكرة جديد (message2) مثلا ، وذلك كالآتى :

```
message2 = "R to return to main menu or  "
```

ويلاحظ كتابة مسافة خالية فى نهاية السلسلة الحرفية . ثم يتم كتابة الأمر التالى :

```
STUFF(message1,6,0,message2)
```

والمعامل الأول لهذه الدالة (message1) هو الرسالة الأصلية . والمعامل الثانى (6) هو رقم الحرف الذى يبدأ عنده ادخال السلسلة الحرفية الثانية (message2). والمعامل الثالث (0) يحدد عدد الحروف التى يتم استبدالها من السلسلة الأولى بحروف أخرى من السلسلة الثانية . والصفر فى هذه الحالة يعنى أنه لايتم استبدال أى حرف . لأن السلسلة الثانية مطلوب اضافتها دون حذف أى حروف من السلسلة الأولى .

والمعامل الرابع هو السلسلة الحرفية الثانية (message2) .

ويمكن تنفيذ نفس هذه العملية بكتابة السطر التالى :

```
SUBSTR(message1,1,5) + message2 + SUBSTR(message1,6)
```

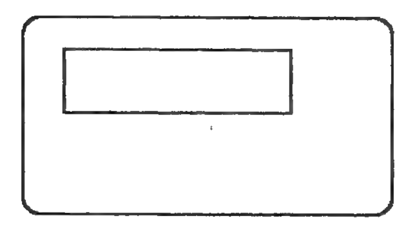
كما يمكن استخدام الأمر STUFF() فى استبدال حروف داخل سلسلة حرفية بحروف أخرى . وكذلك فى مسح حروف داخل سلسلة حرفية وذلك باستبدالها بحروف خالية (Spaces) .

١٩ - ٦ رسم الخطوط حول البيانات

يمكن رسم خطوط مفردة (Single) أو خطوط مزدوجة حول البيانات وذلك باستخدام الأمر @...TO . فمثلا يمكن كتابة السطر التالى :

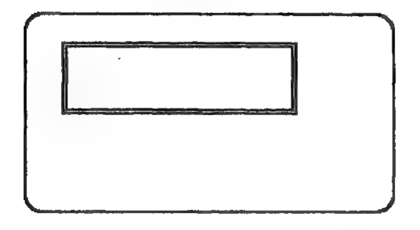
```
@ 2,2 TO 7,50
```

وهذا يؤدى الى رسم مستطيل يبدأ من النقطة (2,2) الى النقطة (7,50) .



كما يمكن كتابة السطر التالى :

```
@ 2,2 TO 7,50 DOUBLE
```



وهذا يؤدى الى رسم مستطيل بخطوط مزدوجة (Double) يبدأ من النقطة (2,2) الى النقطة (7,50) .

ويمكن رسم خط أفقى مزدوج بكتابة السطر التالى :

```
@ 2,2 TO 2,79 Double
```

وذلك بتثبيت رقم السطر .

كما يمكن بالمثل رسم خط رأسي مزدوج بكتابة السطر التالى :

```
@ 2,2 TO 23,2 DOUBLE
```

و ذلك بتثبيت رقم العمود .

ويمكن مسح المستطيل الذى سبق تكوينه باستخدام الأمر @...CLEAR TO وذلك كالآتى :

```
@ 2,2 CLEAR TO 7,50
```

١٩ - ٧ استخدام ملفات الذاكرة

عندما تكون هناك رسائل يتم كتابتها فى كثير من البرامج ، فمن المفيد تخزين هذه الرسائل فى متغيرات ذاكرة ثم تخزين هذه المتغيرات فى ملف ذاكرة (memory file) . وذلك لتوفير الوقت والجهد اللازم لكتابتها فى كل برنامج . فمثلا عندما تكون هناك رسالة كالآتى :

```
Send report to printer?(Y/N)
```

فيمكن تخزين هذه الرسالة فى متغير ذاكرة (message1) . وعندما يراد عرض هذه الرسالة فى مكان معين على الشاشة ، يتم كتابة السطر التالى :

```
@ 5,5 SAY message1
```

كما يمكن تخزين أى خطوط أو أشكال ثم عرضها على الشاشة فى أى مكان .

١٩ - ٨ تكرار الحروف (Repeating Characters)

تستخدم الدالة (REPLICATE()) فى تكرار حرف معين عدة مرات . وتفيد هذه العملية فى رسم أشكال معينة باستخدام هذه الدالة مع الداله CHR() التى تستخدم فى عرض الحروف الخاصة (Special Characters) عن طريق كتابة كود الآسكى الخاص بكل حرف . ويستخدم هذا فى الحصول على أشكال طريفة للشاشة كما سيتم الايضاح فيما بعد .

ويتم استخدام الداله REPLICATE() كالآتى مثلا :

```
@ 4,4 SAY REPLICATE("*",75)
```

ويؤدى هذا الى ظهور الحرف (*) مكررا ٧٥ مرة ابتداء من النقطة (4,4) .

١٩ - ٩ انشاء ملفات التشكيل (Format Files)

كما سبق الايضاح في الجزء الخاص ببرنامج المساعد (Assistant) فان الأوامر (CHANGE, EDIT, APPEND) تؤدى الى ظهور شاشة الادخال حتى يستطيع المستخدم ادخال البيانات أو تعديلها . هذه الشاشة قد تكون هى الشاشة المبدئية (Default) اذا لم يتم انشاء شاشة ادخال أخرى . أو يتم تصميم شاشة ادخال بالطريقة التى سبق شرحها في برنامج المساعد (Assistant) ، ثم فتح الملف الخاص بهذه الشاشة (Format file) باستخدام الأمر (SET FORMAT TO) ثم كتابة اسم الملف . فى هذه الحالة تظهر شاشة الادخال التى تم تصميمها عند استخدام أى امر من الأوامر CHANGE, EDIT, APPEND .

وكما سبق الايضاح أيضا فان تصميم الشاشة يتم عن طريق قائمة الانشاء (CREATE) ، أو كتابة الأمر (CREATE SCREEN) ثم كتابة اسم الملف من مشيرة النقطة (Dot Prompt) . وهذا يؤدى الى عرض حقول ملف قاعدة البيانات ، كما يسمح للمستخدم باختيار الحقول المطلوبة وتحديد الأماكن التى يتم وضع الحقول فيها .

وهذه العملية تؤدى الى انشاء ملفين ، أحدهما يسمى ملف الشاشة (Screen File) والذى يتميز بالامتداد (.SCR) ، والملف الآخر يسمى ملف التشكيل (Format File) ويتميز بالامتداد (.FMT) .

وعندما يراد عمل أى تعديل فى هذه الشاشة باستخدام الأمر (MODIFY SCREEN) ، فان هذا التعديل يتم من خلال ملف الشاشة (Screen File) . وهذه التعديلات تنتقل آليا الى ملف التشكيل (Format File).

وهناك طريقة أخرى لانشاء ملفات التشكيل (Format File) باستخدام ملفات الأوامر (Command Files) . ويتم ذلك عن طريق الأمر (MODIFY COMMAND) ثم كتابة اسم الملف المطلوب انشاؤه مع مراعاة اضافة الامتداد (.FMT) الى هذا الاسم وذلك كالآتى مثلا :

```
MODIFY COMMAND F1.FMT
```

فى هذه الحالة يتم فتح ملف أوامر جديد للكتابة فيه . ويتم كتابة الأوامر التى لاتخرج عن أوامر (@...SAY) وأوامر (@...GET) التى تحدد أماكن الحقول على الشاشة .

فمثلا يمكن كتابة السطور التالية :

```
@    1,9  SAY "Name:"
@    1,20 GET Name
@    3,6  SAY "Address:"
@    3,20 GET Address
@    5,1  SAY "Telephone No"
@    5,20 GET Tel_No
```

ويؤدى هذا الى ظهور الشاشة التالية :

Name :	
Address :	
Telephone No :	

ويلاحظ أن البرنامج لايحتاج الى الأمر (READ) ، وذلك لأن الأوامر(CHANGE, EDIT, APPEND) تتولى تخزين البيانات التى يتم ادخالها . ولكن فى حالة الحاجة الى استخدام عدة شاشات ادخال يستخدم الأمر (READ) فى نهاية كل شاشة .

١٩ - ١٠ استخدام ملف التشكيل

بعد انشاء ملف التشكيل يتم فتح هذا الملف للاستخدام عن طريق الأمر (SET FORMAT TO) ثم كتابة اسم الملف السابق انشاؤه . ويتم ذلك عن طريق كتابة الأوامر التالية مثلا :

```
USE Cadets
SET FORMAT TO F1
APPEND
```

والسطر الأول يؤدى الى فتح ملف قاعدة البيانات (Cadets) .
والسطر الثانى يؤدى الى فتح ملف التشكيل (F1) السابق انشاؤه .
والسطر الثالث يؤدى الى عرض شاشة الادخال لادخال البيانات الى الملف .

ولاغلاق ملف التشكيل يستخدم الأمر (CLOSE FORMAT) أو الأمر (SET FORMAT TO) دون كتابة أى شيء بعده .

ملاحظة

يجب الاهتمام دائما باغلاق جميع ملفات التشكيل السابق فتحها عند الانتهاء من استخدامها . وذلك لأن عدد الملفات المسموح بفتحها من جميع الأنواع يكون محددا .

١٩-١١ استخدام عدة صفحات للادخال (Multiple Pages)

يتيح البرنامج لمخطط البرامج استخدام عدة شاشات ادخال للبيانات . وهذا يساعده على تقسيم البيانات على عدة صفحات حتى يستطيع عرض جميع الحقول خصوصا اذا كانت قاعدة البيانات كبيرة وتحتوى على عدد كبير من الحقول . حيث يتم كتابة الأمر (READ) فى كل مكان يراد مسح الشاشة عنده وعرض شاشة جديدة .

وعندما يقوم المستخدم بادخال البيانات ، فان البرنامج ينتقل الى الشاشة الثانية بمجرد امتلاء الشاشة الأولى . ويمكن للمستخدم فى هذه الحالة الرجوع الى الشاشة السابقة باستخدام مفتاح (PgUp) . كما يمكنه الانتقال الى الشاشة التالية باستخدام المفتاح (PgDn) .

ملاحظة

مهما زادت شاشات الادخال فان عدد أوامر الـ (GET) يجب الا يزيد عن ١٢٨ .

١٩ - ١٢ التعامل مع حقول الملاحظات (Memo Fields)

يتم تخزين حقول الملاحظات فى ملف آخر منفصل عن ملف قاعدة البيانات . لذلك فان التعامل معها يختلف عن التعامل مع أى حقل آخر . فمثلا لايمكن تخزين هذه الحقول فى متغيرات الذاكرة (Memory Variables) . وأيضا لايمكن عرض هذه الحقول باستخدام الأمر (@..SAY) .

وللكتابة فى حقل الملاحظات يتم عرض شاشة التصحيح عن طريق الأمر (EDIT) أو الأمر (CHANGE) ، فتظهر شاشة الادخال . وعندما يراد تعديل حقل الملاحظات ، يتم وضع مؤشر التصحيح على العمود الضوئى الخاص بالملاحظات . ثم بالضغط على مفتاحى (Ctrl-PgDn) يتم فتح حقل الملاحظات لتصحيحه . وعند الانتهاء يتم الضغط على مفتاحى (Ctrl-PgUp) لتخزين الحقل .

ولتنفيذ هذه العملية من خلال البرنامج يتم أولا انشاء ملف تشكيل

(Format File) لحقل الملاحظات الذي يسمى (Notes) مثلا كالآتي :

```
* Notch.fmt - format file for changing memo field
@ 10,10  SAY "Press <Ctrl><PgDn> to edit notes"
@ 12,10  SAY "To save your changes, Press<ctrl><PgUp>"
@ 14,10  SAY "Press<Return>to return back"
@ 16,10  GET Notes
```

ويلاحظ في هذا الملف عرض رسائل للمستخدم لتوضح له الخطوات المطلوب اتباعها لفتح حقل الملاحظات والكتابة فيه ثم التخزين .

ولتعديل حقل الملاحظات يتم كتابة الأوامر التالية في البرنامج :

```
USE Cadets
GOTO recnum
SET FORMAT TO Notch
CHANGE NEXT 1 FIELD NOTES
CLOSE FORMAT
USE
```

والسطر الأول في البرنامج يؤدى الى فتح ملف قاعدة البيانات الذي يسمى (Cadets).

والسطر الثاني يؤدى الى الذهاب الى سجل محدد سبق تخزين رقمه في متغير الذاكرة (Recnum) .

والسطر الثالث يؤدى الى فتح ملف التشكيل (Notch) الخاص بتعديل حقل الملاحظات .

والسطر الرابع يؤدى الى ظهور شاشة الادخال الخاصة بحقل الملاحظات (Notes) . والتي يقوم المستخدم من خلالها بادخال الملاحظات التي يريد ادخالها أو تعديلها اذا كان سبق ادخالها .

والسطر الخامس يؤدى الى اغلاق ملف التشكيل .

والسطر السادس يؤدى الى اغلاق جميع الملفات .

ويمكن عرض محتويات حقول الملاحظات على الشاشة باستخدام الأمر (DISPLAY) والأمر (LIST) . كما يمكن استخدام الأمر (?) أيضا لنفس

الغرض . كما يمكن التحكم فى عرض الملاحظات المعروضة عن طريق كتابة الأمر التالى :

```
SET MEMOWIDTH TO
```

ثم كتابة العرض المطلوب استخدامه .

فعندما يراد مثلا عرض الملاحظات بعرض ٦٠ حرفا ، يتم كتابة الأمر التالى :

```
SET MEMOWIDTH TO   60
```

فى هذه الحالة يتم عرض الملاحظات فى سطور كل سطر منها طوله ٦٠ حرفا .

ويمكن كتابة هذا الأمر فى ملف المواصفات (Config.sys) .

١٩-١٣ زيادة مخزن الكتابة المؤقت (TYPEAHEAD BUFFER)

عندما يقوم المستخدم بادخال البيانات فان الحروف التى يكتبها تخزن فى مخزن ذاكرة مؤقت (Buffer) . و عندما يصل عدد الحروف الى عدد محدد ، تنتقل هذه الحروف الى الملف . و العدد المبدئى (Default) لهذه الحروف هو ٢٠ حرفا . كما يمكن زيادة هذا العدد عن طريق الأمر (SET TYPEAHEAD TO) ثم كتابة أى عدد من صفر الى ٣٢ ألف حسب سعة الذاكرة المتاحة . وكلما كان هذا العدد كبيرا ، ساعد ذلك على ادخال البيانات أسرع . حيث أن ذلك يتيح للمستخدم الكتابة بسرعة أثناء ادخال البيانات .

ملاحظة

الأمر (SET TYPEAHEAD) لا يعمل الا فى حالة (SET ESCAPE ON) . لذلك لايفضل استخدامه الا فى حالات الضرورة حتى يمكن استخدام الأمر (SET ESCAPE OFF) .

ملاحظة

ما سبق ذكره فى هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، FoxBase + ، FoxPro .

الباب العشرون

اختبار مدخلات المستخدم

عند استقبال أى مدخلات للمستخدم فانه من الطبيعي والمتوقع أن يخطيء المستخدم ويكتب حروفا قد تكون غير مطلوبة . وفى هذه الحالة قد يؤدى ادخال هذه الحروف الى توقف البرنامج أو الى عدم الحصول على المخرجات المطلوبة . ولذلك فمن المفيد التحكم فى مدخلات المستخدم وعدم قبول أى مدخلات غير مسموح بها . ويتم ذلك عن طريق عمل مايشبه الترشيح (Filtering) لهذه المدخلات بحيث لايدخل الى البرنامج الا المدخلات الصحيحة ، أما أى مدخلات أخرى فانها لاتمر من هذا المرشح . ويمكن أن يتم ذلك بعدة طرق يتم القاء الضوء عليها فى هذا الباب .

٢٠ - ١ استخدام الاختيارات العددية (Numeric Choices)

يفضل عند عرض قائمة اختيارات (Menu) للمستخدم ، أن تستخدم الأرقام فى القائمة ويقوم المستخدم باختيار رقم من هذه القائمة . حيث أن الأرقام يمكن التحكم فيها عن طريق تحديد مدى معين . فمثلا اذا كانت هناك قائمة تحتوى على الاختيارات التالية :

```
1 -  Add new records
2 -  Edit
3 -  Delete
4 -  Display
5 -  Return
```

فى هذه الحالة يمكن كتابة السطور التالية :

```
Choice = 1
@ 5,5 GET Choice PICTURE '9' RANGE 1,5
READ
```

ويتم التحكم فى الرقم الذى يقوم المستخدم بادخاله عن طريق تحديد مدى لهذا الرقم من (1) الى (5) . وفى حالة ادخال المستخدم لأى رقم لا يقع فى هذا المدى فان البرنامج لايقبله . ولاينتقل البرنامج الى الخطوات التالية الا بعد ضغط المستخدم على الرقم الصحيح .

٢٠ - ٢ توقع احتمالات الخطأ

يمكن بعد أن يقوم المستخدم بادخال البيانات اعطاؤه الفرصة لاختبار هذه المدخلات والتأكد من صحتها . وذلك عن طريق عرض هذه البيانات للمستخدم ثم عرض الرسالة التالية مثلا :

```
Is this correct ? (Y/N)
```

ثم ينفذ البرنامج حلقة تكرارية عن طريق الأمر (DO WHILE) تساعد المستخدم على تصحيح البيانات التى قام بادخالها . وذلك فى حالة كتابة المستخدم (N) بما يفيد عدم صحة البيانات . أما فى حالة صحة البيانات فيتم اكمال تنفيذ البرنامج .

ولكن هناك احتمال أن يكتب المستخدم الحرف (Y) كبيرا (Capital) او صغيرا (Small) . وهذا يمكن التغلب عليه عن طريق استخدام الرمز (!) فى الصورة المطلوبة (PICTURE) . وذلك عن طريق السطور التالية مثلا :

```
STORE ' ' TO answer
@ 5,0 GET answer PICTURE '!'
READ
```

فى هذه الحالة اذا أدخل المستخدم (y) أو (n) يقوم البرنامج بتحويلها الى حروف كبيرة (Capital) . ولكن ماذا لو أدخل المستخدم أى حرف آخر غير (y) أو (n) ؟

فى هذه الحالة يجب استبعاد أى حرف آخر يدخله المستخدم غير الحرفين Y أو N . ويستخدم لذلك المعامل $ والذى يعنى وجود حرف معين ضمن سلسلة حروف (String).

فمثلا عند كتابة الأمر التالى :

```
DO WHILE .NOT. answer $ 'YN'
```

فان هذا يؤدى الى العودة دائما الى الحلقة التكرارية فى حالة ادخال المستخدم لأى حرف غير (Y) أو (N) .

ولتوضيح ذلك يمكن دراسة مجموعة السطور التالية :

```
choice = ' '
     DO WHILE .NOT. choice $ 'YN'
     choice = ' '
     @ 15,15 GET choice PICTURE '!'
     READ
ENDDO
```

والسطر الأول يؤدى الى انشاء متغير الذاكرة (choice) .

والسطر الثانى يؤدى الى دخول الحلقة التكرارية فى حالة ادخال المستخدم لأى حرف آخر غير حرفى (Y) أو (N) . وفى حالة ادخال

المستخدم للحرف (Y) أو (N) فان الحلقة التكرارية لايتم تنفيذها ، ويتم تنفيذ باقى أوامر البرنامج بناء على ذلك . أما اذا أدخل المستخدم (y) صغيرة (small) أو (n) صغيرة . فان الحلقة التكرارية تنفذ مرة واحدة . لأن هذا الحرف يتم تحويله الى حرف كبير (Capital) من خلال السطر الرابع عن طريق الصورة (PICTURE) . وفي هذه الحالة يتم تنفيذ باقي أوامر البرنامج بناء على اختيار المستخدم اذا كان (y) أو (n) .

والسطر الثالث يؤدى الى مسح محتويات متغير الذاكرة (choice) في حالة ادخال المستخدم لأي حرف غير (y) أو (n) . فمثلا عند ادخال أى حرف آخر مثل S مثلا ، فان البرنامج ينفذ الحلقة التكرارية ، ويصل الى السطر الخامس حيث يطلب من المستخدم ادخال حرف . وفى نفس الوقت يجد المستخدم الحرف (S) مكتوبا في العمود الضوئي (Highlight) . وربما يسبب له ذلك شيئا من الارتباك . لذلك يستخدم السطر الثالث في مسح محتويات العمود الضوئي (Highlight) الممثل للمتغير (Choice) حتى يظهر أمام المستخدم خاليا . وذلك يعنى أن البرنامج لم يتقبل الحرف الذى تم ادخاله .

ملاحظة

يجب ملاحظة الفرق بين المعامل $ والدالة ()SUBSTR. فالمعامل $ يبحث عن حروف معينة فى سلسلة حرفية واذا وجدها فانه يعطى القيمة .T. أى صحيح (True). واذا لم يجدها يعطى .F. أي غير صحيح (False) . بينما الدالة ()SUBSTR تعطى جزءا من السلسلة الحرفية (String) .

كما يمكن دراسة المثال التالى لتوضيح طريقة أخرى لاختبار مدخلات المستخدم .

```
name = SPACE(30)
DO WHILE .T.
     @ 5,5 GET name
     READ
     IF name <> SPACE(30)
        EXIT
     ENDIF
     @ 10,10 SAY "No blank name allowed"
ENDDO
```

وهـــذا البرنامـــج يستخـــدم مـــا يسمـــى بالطـــريقة السلبيـــة (negative approach). وهو يعنى الخروج من الحلقة التكرارية فى حالة ادخال المستخدم القيمة الصحيحة. فاذا أدخل المستخدم أى حروف فى المتغير (Name) ، فان الشرط الموجود بعد (IF) يتحقق . وبالتالى يتم تنفيذ الأمر التالى وهو(EXIT) الذى يؤدى الى الخروج من الحلقة التكرارية ، وبالتالى تنفيذ باقى أوامر البرنامج .

أما اذا لم يدخل المستخدم أى حروف ، فان المتغير name يظل خاليا أى محتويا على SPACE(30) . وذلك يؤدى الى عدم تحقق الشرط بعد (IF) ، وبالتالى الاستمرار فى تنفيذ الحلقة التكرارية حتى يقوم المستخدم بادخال أى حروف فى المتغير (name) .

كما يمكن استخدام الطريقة الايجابية (Positive approach) . وذلك بكتابة الأمر بعد (IF) كالآتى :

```
IF name = SPACE(30)
```

وهناك طرق متعددة لاختبار مدخلات المستخدم . ويمكن لمخطط البرامج اختيار الطرق المناسبة . ولكن المهم أن يغطى جميع احتمالات الخطأ .

٢٠ - ٣ استخدام الداله INKEY()

يمكن استخدام الداله INKEY() فى اختبار مدخلات المستخدم . وهذه الدالة تعطى القيمة العددية الممثلة لكود الآسكى (ASCII Code) الخاص بآخر مفتاح تم الضغط عليه بواسطة المستخدم . ولكى يتم توضيح ذلك يمكن دراسة الأوامر التالية :

```
i = o
DO WHILE i = o
     i = INKEY()
ENDDO
? i
```

وهذه الأوامر تؤدى الى استمرار تنفيذ الحلقة التكرارية حتى يضغط المستخدم على أى مفتاح . فمثلا اذا ضغط المستخدم على المفتاح (A) يترك البرنامج الحلقة التكرارية ويعرض الرقم (65) الذى يمثل كود الآسكى الخاص بالحرف (A) .

ويجب ملاحظة أن الداله INKEY() لاتعمل مع مفتاح ALT .حيث تعطى القيمة صفر . وكذلك الضغط على مفتاح (ALT) مع أى مفتاح آخر يؤدى الى نفس النتيجة .

ويمكن دراسة مجموعة الأوامر التالية للتعرف على استخدام الدالة INKEY() فى اختبار مدخلات المستخدم .

```
i = o
DO WHILE i = o
     i = INKEY()
     IF UPPER(CHR(i)) $ "ABCDEFGHIJKLX"
          EXIT
     ENDIF
          i = o
ENDDO
```

وتؤدى هذه الأوامر الى الدخول فى حلقة تكرارية طالما كانت قيمة المتغير (i) تساوى صفرا . وعندما يضغط المستخدم على أى مفتاح فان المتغير (i) يحتوى على القيمة العددية الممثلة لكود الآسكى الخاص بهذا المفتاح .

والأمر (IF) يختبر هذه القيمة العددية بعد تحويلها الى الحرف المقابل باستخدام الداله CHR() وبعد تحويله الى حرف كبير (CAPITAL) .

فاذا كان هذا الحرف ضمن الحروف الواضحة فى الشرط ، يخرج البرنامج من الحلقة التكرارية وينتقل الى أوامر البرنامج التالية . حيث ينفذ الأوامر التى تختص بكل حرف من هذه الحروف حسب اختيار المستخدم .

٢٠ - ٤ الضغط على مفتاح الادخال

فى بعض الأحيان يراد استخدام مفتاح الادخال كأحد الاختيارات فى البرنامج . فى هذه الحالة يتم اختبار السلسلة الحرفية (String) التى يدخلها المستخدم . فاذا كان طولها = صفر ينفذ البرنامج أوامر معينة . ويمكن توضيح ذلك من الأوامر التالية :

```
IF LEN(TRIM(choice)) = 0
     DO something
ELSE
     DO another
ENDIF
```

حيث أن المتغير (choice) هو متغير يتم انشاؤه من خلال البرنامج واعطاؤه القيمة (space) ، أى أنه سلسلة حرفية خالية .

والشرط بعد (IF) يؤدى الى اختبار طول السلسلة الحرفية الموجودة فى المتغير (choice) ، وذلك بعد حذف المسافات الزائدة باستخدام الدالة (TRIM) .

فاذا ضغط المستخدم على مفتاح الادخال دون كتابة أى حروف يصبح المتغير (choice) خاليا، أى محتويا على فراغ (space) فقط . كما يؤدى الأمر (TRIM) الى حذف هذا الفراغ وبالتالى يصبح طول المتغير الحرفى صفرا . وهذا يوضح أهمية الأمر (TRIM) فى هذه الحالة . حيث ان السلسلة الحرفية المحتوية على فراغ (space) لايكون طولها صفرا .

٢٠ - ٥ اختبار مسطرة المسافات (Space Bar)

كما سبق الايضاح ، يمكن حذف المسافات الخالية فى أول السلسلة الحرفية أو فى آخرها باستخدام الدوال LTRIM() ، RTRIM() ، TRIM() .

ولكن ماذا لو أدخل المستخدم مسافة خالية داخل السلسلة الحرفية ؟ .

فى هذه الحالة لاتصلح هذه الدوال لاكتشاف الخطأ الذى أدخله المستخدم . ولكن يمكن استخدام الدالة AT() . هذه الدالة تختبر وجود حروف معينة داخل سلسلة حرفية . فاذا وجدت هذه الحروف فانها تعطى مكان هذه الحـــروف . واذا لم تجدها فانها تعطى القيمة صفر . ولتوضيح ذلك يمكن دراسة الأوامر التالية :

```
DO WHILE .T.
       ACCEPT "What is the account number?" TO choice
       STORE LTRIM(TRIM(choice)) TO choice
       IF AT(" ",choice) > 0
            CLEAR
            ? CHR(7)
            @ 10,10 SAY "You typed space" +;
            "in the account number"
            ?
            WAIT "Press any key to try again"
            RELEASE choice
            LOOP
       ENDIF
       EXIT
  ENDDO (WHILE .T.)
```

وفى هذا البرنامج ، عندما يدخل المستخدم رقم حساب معين ، فان الدالة LTRIM() والداله TRIM() تؤديان الى حذف الفراغات (Spaces) من أول وآخر

هذا الرقم . ثم تقوم الداله ()AT باختبار وجود فراغات داخل هذا الرقم . فاذا وجدت أى فراغ فانها تعطي الرقم الدال على مكان هذا الفراغ فى السلسلة الحرفية (Choice). أى تعطى قيمة أكبر من صفر . وفى هذه الحالة يتحقق الشرط بعد (IF) ويتم تنفيذ الأوامر التالية . حيث يتم مسح الشاشة وتشغيل جرس التحذير (Bell) ثم عرض الرسالة التي توضح للمستخدم الخطأ الذى وقع فيه . كما يؤدى الأمر (WAIT) الى الانتظار (Pause) حتى يقرأ المستخدم الرسالة ثم يضغط على أى مفتاح بعد ذلك . ثم يتم مسح محتويات المتغير (choice) باستخدام الأمر (RELEASE) . ويلى ذلك الأمر (LOOP) الذى يؤدى الى الرجوع الى أول الحلقة التكرارية .

وفى حالة عدم وجود مسافات داخل السلسلة الحرفية بعد (IF)، فان البرنامج يتخطى الاوامر بعد (IF) ثم ينفذ الأمر (EXIT) ليخرج من الحلقة التكرارية بعد تخزين رقم الحساب في المتغير (choice) .

ويجب ملاحظة استخدام الأمر (?) دون كتابة أى شيء حيث يؤدي ذلك الى ترك سطر خال . وذلك لعرض الرسالة مع ترك سطر خال بينها وبين الرسالة السابقة .

٢٠ - ٦ اختبار نوع المدخلات

هناك طرق أخرى لاختبار مدخلات المستخدم عن طريق اختبار نوع هذه المدخلات اذا كان حرفيا (Character) أو عدديا (Numeric) أو تاريخيا (Date) أو منطقيا (Logical). ويتم ذلك باستخدام الدوال (ISALPHA()) ، (ISLOWER()) ، (ISUPPER()) . وهى تعطى القيمة المنطقية صحيح (True) أو غير صحيح (False). وهذه الدوال تختبر أول حرف فقط في المدخلات . ويمكن استخدام هذه الدوال فى اكتشاف أخطاء المستخدم . فمثلا يمكن كتابة البرنامج الفرعى التالى :

```
@ 10,10 GET choice
READ
IF ISALPHA(choice)
    DO something
ELSE
    DO error
ENDIF
```

حيث (error) هو برنامج فرعي آخر يؤدى الى اظهار رسالة للمستخدم توضح له الخطأ . بالاضافة الى تنفيذ بعض العمليات الأخرى .

وهناك دالة أخرى (Type()) تستخدم لتحديد نوع الدخلات . وهى لاتعطى

قيمة منطقية (صحيح أو غير صحيح) ولكنها تعطي أول حرف يمثل نوع المدخلات مثل (C) للمدخلات الحرفية (Character) ، أو (N) للمدخلات العددية (Numeric) ، أو (D) للمدخلات التاريخية ، أو (M) للملاحظات (Memo) ، أو (L) للمدخلات المنطقية (Logical) . وفي حالة ادخال حروف غير مطابقة لأي نوع من هذه المدخلات ، أو عند ادخال حروف في متغير ذاكرة دون انشاء هذا المتغير أولا ، فان هذه الدالة تعطي الحرف (U) الذي يعني غير معرف (Undefined).

والأوامر التالية توضح استخدام هذه الدالة .

```
STORE "Hello" TO message1
? TYPE('message1')
```

وعند الضغط على مفتاح الادخال يظهر الحرف (C) أي (Character) .

٢٠ - ٧ استخدام الأمر (ON)

يستخدم هذا الأمر للتفرع الى برنامج فرعي آخر بناء على ضغط المستخدم على مفتاح الهروب (ESC) أو أي مفتاح آخر ، أو عند حدوث خطأ معين في البرنامج (Error).

فمثلا الأمر (ON ESCAPE) يستخدم لاختبار ضغط المستخدم على مفتاح الهروب (ESC) خلال تنفيذ البرنامج . فاذا ضغط المستخدم على مفتاح الهروب أثناء تشغيل البرنامج ، يتفرع البرنامج الى برنامج فرعي آخر . فمثلا يمكن كتابة الأمر التالي :

```
ON ESCAPE DO warning
```

وهذا يؤدي الى التفرع الى برنامج (Warning) عند ضغط المستخدم على مفتاح الهروب .

ويمكن كتابة هذا الأمر في بداية البرنامج الرئيسي ، حيث يصبح مؤثرا في جميع البرامج الفرعية .

والأمر (ON KEY) يؤدي نفس الشيء ، ولكن عند ضغط المستخدم على أي حرف .

اما الأمر (ON ERROR) فانه يؤدي الى التفرع الى برنامج فرعي عند حدوث أي خطأ في تشغيل البرنامج . حيث يمكن من خلال هذا البرنامج الفرعي عرض رسائل

خطأ للمستخدم ، وعدم الاعتماد على رسائل الخطأ البدئية (Default) الموجودة في برنامج (+ DBase III) .

ملاحظة

ما سبق ذكره في هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، + FoxBase ، FoxPro .

الباب الحادي والعشرون

التعامل مع قاعدة البيانات

عند تصميم البرنامج بواسطة احد برامج عائلة (DBase)، فان هذا البرنامج لايتعامل مع البيانات التى يدخلها المستخدم عن طريق لوحة المفاتيح فقط ، ولكنه يتعامل أيضا مع البيانات المخزنة فى قاعدة البيانات . وهذا الباب يشرح تعامل البرنامج مع ملف قاعدة البيانات من حيث تصميم هذا الملف والملفات الملحقة به مثل ملف الفهرس (Index File) أو ملف البحث (Query File) أو .. الخ . وكذلك من حيث فتح هذا الملف والملفات الملحقة به .

٢١ - ١ تصميم قاعدة البيانات

عادة يتم تصميم قاعدة البيانات خارج البرنامج ، أى عن طريق برنامج المساعد (Assistant) أو من خلال مشيرة النقطة (Dot Prompt) بأمر منفصل عن أوامر البرنامج . ويستخدم لذلك الأمر (CREATE) ثم كتابة اسم الملف المطلوب انشاؤه . كما يستخدم الأمر (MODIFY STRUCTURE) في انشاء الملف أيضا بالاضافة الى تعديله .

وملف قاعدة البيانات يستطيع تخزين حتى بليون سجل . وكل سجل يحتوى على مايقرب من ٤٠٠٠ حرف موزعين على عدد من الحقول لايزيد عن ١٢٨ حقلا (Field) . ولكن يجب ملاحظة أنه عند زيادة حجم ملف قاعدة البيانات بدرجة كبيرة، فان ذلك يؤدى الى بطء معالجة البيانات المخزنة ، وبالتالى يؤثر على كفاءة البرنامج .

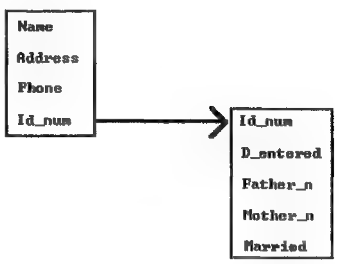
ولعلاج ذلك يمكن تقسيم ملف قاعدة البيانات الى مجموعة ملفات ، مع تحديد حقل مشترك بينها ، مع ملاحظة أن هذا الحقل يجب أن يكون منفردا (Unique) ، أى يعطى بيانا محددا لكل سجل فى الملف ، بحيث لايكون هناك سجلان مشتركان فى هذا البيان .

وبرنامج (DBase III+) يسمح بتصميم أى عدد من الملفات المرتبطة (Related) دون أى تحديد (Limitation) لهذا العدد . ولكن عند فتح هذه الملفات، فهناك حد أقصى لعدد الملفات المفتوحة متضمنا ملفات قاعدة البيانات والملفات الملحقة بها مثل الملفات الفهرسية (Index Files) ، وملفات البحث (Query Files) و .. الخ .

وهذا العدد من الملفات يتم تحديده في ملف المواصفات (Config.sys) ويرتبط بحجم الذاكرة المؤقتة المتاح .

والقاعدة العامة في استخدام الملفات المتعددة المرتبطة (Related) هى استخدام حقل واحد مشترك ومنفرد (Unique) كما سبق الايضاح . ويجب تجنب استخدام حقول أخرى مشتركة بين الملفات حتى لاتؤدى الى بطء تشغيل البرنامج أو استهلاك المساحة التخزينية المتاحة على القرص .

والشكل التالى يوضح ملفين مرتبطين عن طريق حقل رقم تحقيق الشخصية .



شكل (٢١ - ١)

ويمكن عن طريق استخدام ملفات متعددة في قاعدة البيانات عزل البيانات السرية التي يكون مطلوبا عدم التعامل معها بواسطة أشخاص معينين . وذلك عن طريق استخدام ملف منفصل لهذه البيانات السرية ، واستخدام كلمة المرور (Password) في تحديد الأشخاص المسموح لهم بالتعامل مع هذا الملف . وهذا يعتبر ضروريا عند استخدام البرنامج فى شبكات الحاسب (Networks) .

٢١ - ٢ هيكل ملف قاعدة البيانات

يتم تحديد هيكل ملف قاعدة البيانات باستخدام الأمر (CREATE) أو الأمر (MODIFY STRUCTURE) كما سبق الايضاح . وفى هذه الحالة يتم فتح هيكل خال للملف لتحديد أسماء الحقول ونوعها وعرضها .

٢١ - ٢ - ١ تحديد أسماء الحقول

أسماء الحقول (Field Names) يصل طولها الى عشرة حروف ، ويجب أن تبدأ بحرف ولا تحتوى على مسافات خالية (Spaces) . ويمكن استخدام الحروف والأرقام والشرطة السفلية (Underscore) .

٢١ - ٢ - ٢ تحديد أنواع الحقول

يتم تحديد أنواع الحقول (Field Types) عن طريق كتابة الحرف الأول من كل نوع . وذلك كالآتى :

C وتعنى حقل حرفي (أول حرف من Character)
D وتعنى حقل تاريخي (أول حرف من Date)
L وتعنى حقل منطقي (أول حرف من Logical)
M وتعنى حقل ملاحظات (أول حرف من Memo)
N وتعنى حقل عددى (أول حرف من Numeric)

كما يمكن تغيير نوع الحقل بالضغط على مسطرة المسافات (Space Bar) حيث تؤدى كل ضغطة عليها الى التحويل من نوع الى آخر .

وفي الأجزاء التالية يتم شرح كل نوع من الحقول بالتفصيل .

٢١ - ٢ - ٢ - ١ الحقول الحرفية

وهى حقول يمكن استخدامها في ادخال أى حروف من لوحة المفاتيح وهى تشمل الحروف والأعداد والحروف الخاصة (Special Characters) والمسافات (Spaces) . وأقصى عرض أو حجم لهذه الحقول هو ٢٥٤ حرفا .

٢١ - ٢ - ٢ - ٢ الحقول التاريخية

وهى حقول تستخدم في تخزين التواريخ . ويمكن عرض التواريخ بعدة صور كما سبق الايضاح . والصورة المبدئية (Default) للتاريخ هى الصورة الأمريكية (mm/dd/yy) . وحقل التاريخ دائما عرضه ٨ حروف حتى في حالة استخدام ٤ حروف ممثلة للسنة (yyyy) باستخدام الأمر (SET CENTURY ON) .

ويمكن اجراء عمليات حسابية على هذه الحقول باضافة عدد من الأيام الى تاريخ معين أو طرح عدد من الأيام من تاريخ معين ، أو طرح تاريخ من تاريخ .. وهكذا .

٢١ - ٢ - ٢ - ٣ الحقول العددية

وهى الحقول التى يتم فيها ادخال الأعداد التى يتم اجراء عمليات حسابية عليها . والحقل العددى يمكن ان يحتوى على ١٥ رقما بما فيها الأرقام العشرية (Decimal) التى يجب ألا تزيد عن ٩ أرقام .

٢١ - ٢ - ٢ - ٤ الحقول المنطقية

وهى حقول لاتقبل الا حرفا واحدا يمثل حالة هذا البيان ، اذا كان

صحيحا (True) أو غير صحيح (False) . فاذا كان صحيحا يتم ادخال أحد الحروف الآتية (T,t,Y,y) ، واذا كان غير صحيح يتم ادخال أحد الحروف الآتية (F,f,N,n) .

٢١ - ٢ - ٢ - ٥ حقول الملاحظات

وهى حقول يتم تصميمها لادخال كميات كبيرة من المعلومات عن كل سجل . ويتم تخزين هذه المعلومات في ملف مساعد يحمل نفس اسم ملف قاعدة البيانات ولكن بالامتداد (.dbt) بدلا من الامتداد (.dbf). ويتم تمييز هذا الحقل فى ملف قاعدة البيانات بكلمة (memo) وعرضه عشرة حروف .

وحقل الملاحظات حجمه الحقيقى متغير تبعا للمعلومات التي يتم تخزينها فيه . ففى حالة عدم ادخال أى بيانات يكون حجمه صفرا . ويمكن ادخال حتى ٥٠٠٠ حرفا فى هذا الحقل . كما يمكن زيادة حجمه عن ذلك كثيرا باستخدام أى برنامج معالجة كلمات آخر غير معالج الكلمات المستخدم فى برنامج (DBase III+) . ويلزم لذلك ادخال اسم هذا البرنامج فى ملف المواصفات (Config.sys) الخاص ببرنامج (DBase III+).

٢١ - ٢ - ٣ تحديد عرض الحقل

عرض الحقل (Field Width) هو أكبر عدد من الحروف والأرقام يمكن كتابته فى الحقل . وفى حالة الحقول العددية ، يتم حساب نقطة الكسر العشرى (Decimal Point) والفاصلة (Comma) ضمن حروف الحقل . كما أن الحقول التاريخية والمنطقية وحقول الملاحظات لها عرض ثابت .

٢١ - ٢ - ٤ فتح ملف قاعدة البيانات

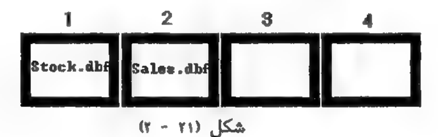
يتم فتح ملف قاعدة البيانات باستخدام الأمر (USE) يليه اسم الملف المطلوب فتحه . ويمكن فتح ملفات قاعدة البيانات وملفات الفهرس الملحقة بها فى بداية البرنامج ، كما يمكن فتح الملفات المطلوبة وقت الحاجة اليها . وهذا يتوقف على حجم البرنامج وعدد الملفات المستخدمة . ولكن يفضل فى جميع الأحوال فتحها وقت الاحتياج اليها فقط للمحافظة على تكامل قاعدة البيانات (Integrity) . حيث أن ترك الملفات مفتوحة مدة طويلة قد يؤدى الى حدوث مشاكل فى هذه الملفات فى حالة قطع التيار الكهربى أو الاغلاق المفاجئ للجهاز . كما يفضل اغلاق الملف بمجرد انتهاء الحاجة اليه .

وفى حالة استخدام عدة ملفات مرتبطة (Related) في نفس الوقت ، يتم فتح هذه الملفات فى مناطق عمل (Work Areas) مختلفة باستخدام الأمر (SELECT) .

فمثلا مجموعة الأوامر التالية تؤدى الى فتح ملفين فى منطقتين للعمل (1 , 2) .

```
SELECT 1
USE Stock INDEX Cust_no
SELECT 2
USE Sales INDEX Sale_no
```

ويلاحظ فتح كل ملف بالاضافة الى فتح الملف الفهرسي المرتبط به من خلال نفس الأمر . والشكل التالى يوضح عملية تخصيص منطقة العمل (Work Area) لكل ملف .



شكل (٢١ - ٢)

ويمكن فتح حتي عشرة مناطق عمل (Work Areas) ، واستخدامها فى فتح عدة ملفات فى نفس الوقت .

ويمكن اغلاق أي ملف وبالتالي اغلاق منطقة العمل الخاصة به . وذلك باستخدام الأمر (USE) دون كتابة أي شيء بعده ، كما يمكن اغلاق جميع ملفات قاعدة البيانات باستخدام الأمر (CLOSE DATABASES) .

٢١ - ٣ استخدام المرادفات (Aliases)

عندما يراد استخدام عدة ملفات قاعدة بيانات من خلال مناطق عمل مختلفة (Work Areas) ، يمكن اختيار مرادفات (Aliases) لاسم كل ملف . وهذه المرادفات توفر علي مخطط البرامج كتابة اسم الملف وملف الفهرس الملحق به (Index File) فى كل مرة يراد فيها اختيار ملف فى منطقة عمل معينة (Work Area) .

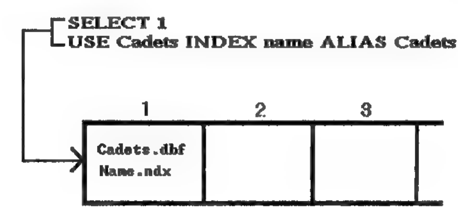
فمثلا عند كتابة الأوامر التالية :

```
SELECT 1
USE Cadets Index Name Alias Cadets
```

وفى كل مرة يراد فتح ملف الطلبة (Cadets) وملف الفهرس الملحق به ، يستخدم الأمر التالى :

```
SELECT Cadets
```

وهذه المرادفات (Aliases) تساعد على توضيح أى قاعدة بيانات مفتوحة فى أى منطقة عمل .



شكل (٢١ - ٢)

تحـذير

يجـب عدم استخدام الحروف من (A) الى (J) منفصلة فى المـرادفـات (Aliases) . لأن هذه الحروف تمثل مناطق العمل من ١ الى ١٠ . ولكن يـمكـن استخدام هذه الحروف مع حروف أخرى أو أعداد أخرى مثـل A1 ، A2 ، A3 ، .. الخ .

٢١ - ٤ انشاء ملف الفهرس (Index File)

كما سبق الايضاح فان الملف الفهرسى (Index File) هو ملف يحتوى على حقلين فقط أحدهما يحتوى على رقم السجل ، والآخر يحتوى على البيانات المطلوب ترتيب ملف قاعدة البيانات بناء عليها . وعن طريق هذا الملف يمكن الوصول بسهولة الى أى سجل فى ملف قاعدة البيانات عن طريق البيانات الموجودة فى الحقل الفهرسى (Key Field) . ولانشاء ملف الفهرس يستخدم الأمر (INDEX) مع كتابة اسم

الحقل الفهرسى وكذلك اسم الملف المطلوب انشاؤه . وذلك كالآتى مثلا :

```
INDEX ON name TO Name
```

وهذا يعنى انشاء ملف فهرسى اسمه Name.ndx بناء على حقل الاسم name كحقل فهرسى .

ويمكن الترتيب بناء على أكثر من حقل . وهذا يساعد على ترتيب السجلات التى تشترك فى الحقل الفهرسى الأول بناء على حقل آخر . فمثلا اذا كان هناك حقل يمثل الاسم الأول (First Name) وحقل يمثل الاسم الثانى (Second Name) فان البرنامج يبدأ بترتيب قاعدة البيانات بناء على حقل الاسم الأول . فاذا كان هناك عدة سجلات تشترك فى الاسم الأول وليكن (Mohamed) مثلا وذلك كالآتى :

```
Mohamed Maged
Mohamed Hasan
Mohamed Tarek
```

فى هذه الحالة يتم ترتيب هذه السجلات بناء على الاسم الثانى (Second Name) وذلك كالآتى :

```
MOHAMED Hasan
Mohamed Maged
Mohamed Tarek
```

ويلاحظ هنا اعادة ترتيب السجلات بناء على الترتيب الهجائى للاسم الثانى .

ولكى يتم انشاء ملف فهرسى (Index File) بناء على حقلين أو أكثر تستخدم علامة الجمع (+) ، وذلك كالآتى :

```
USE Cadets
INDEX ON F_name + L_name TO Name
```

ويجب ملاحظة أن الترتيب يتم تصاعديا حسب الترتيب الهجائى للحروف .

وعند الترتيب بناء على عدة حقول ، يجب التأكد أولا أن هذه الحقول لها نفس النوع (Type) . واذا كان النوع مختلفا فيلزم فى هذه الحالة اجراء عمليات تحويل من حالة الى أخرى .

فمثلا اذا أريد ترتيب بيانات ملف الطلبة (Cadets) بناء على حقل الاسم الأول (F_name) وحقل الاسم الثانى (S_name) وحقل تاريخ الدخول (D_enter)

، فى هذه الحالة يلزم تحويل حقل التاريخ الى حروف . حيث يتم عمل تحويل منفصل للجزء الممثل للسنة والجزء الممثل للشهر والجزء الممثل لليوم . وذلك كالآتى مثلا :

```
USE Cadets
INDEX ON F_name + S_name + STR(YEAR(D_Enter),4)  ; +
STR(MONTH(D_Enter),2) + STR(DAY(D_Enter),2)
```

٢١ - ٥ فتح ملف الفهرس

يتم فتح ملف الفهرس (Index File) باستخدام نفس الأمر (USE) المستخدم فى فتح ملف قاعدة البيانات . وذلك كالآتى مثلا :

```
USE Cadets INDEX Name
```

وهناك طريقة أخرى لفتح ملف الفهرس باستخدام الأمر (SET INDEX TO) ثم كتابة اسم الملف الفهرسى . وذلك كالآتى :

```
SET INDEX TO Name, Class
```

ويجدر العلم أنه عند فتح ملفات الفهرس ، فان أى تعديل أو اضافة للبيانات تؤدى الى تعديل ملف الفهرس أيضا . أى أن أى سجلات تضاف الى قاعدة البيانات يتم ادخالها فى نفس الوقت فى ملف الفهرس .

ويمكن تغيير ترتيب ملفات الفهرس التى سبق فتحها باستخدام الأمر (SET ORDER TO) ثم كتابة رقم ملف الفهرس المراد استخدامه كملف فهرسى رئيسى (Master) . وذلك كالآتى مثلا :

```
SET ORDER TO 2
```

فى هذه الحالة يصبح الملف (Class) هو ملف الفهرس الرئيسى (Master).

ويمكن اعادة الملف الى الحالة غير المفهرسة (Unindexed) وذلك باستخدام الأمر (SET ORDER TO 0) . وهذا يؤدى الى التعامل مع السجلات بترتيبها الأصلى الذى أدخلت به . وذلك مع عدم اغلاق ملفات الفهرس .

ولاغلاق ملفات الفهرس دون اغلاق ملف قاعدة البيانات يستخدم الأمر التالى :

CLOSE INDEX

كما يمكن اغلاقها أيضا باستخدام الأمر التالى :

SET INDEX TO

وذلك دون كتابة أى شيء بعده .

ملاحـظة

رغم أن عدد ملفات الفهرس التى يمكن انشاؤها لملف قاعدة بيانات واحد غير محدود ، حيث يمكن انشاء أى عدد من ملفات الفهرس لكل ملف قاعدة بيانات ، الا أن عدد ملفات الفهرس التى يمكن فتحها فى نفس الوقت لا يزيد عن سبعة ملفات . وهـذا يعتبر كافيا جدا ، اذا أخذنا فى الاعتبار أيضا أن الحقل الفهرسى فى كل ملف يمكن أن يحتوى على أكثر من حقل من حقول ملف قاعدة البيانات .

ويفضل دائما فتح جميع ملفات الفهرس المطلوبة مرة واحدة ، ثم تعديل ترتيب هذه الملفات فى أى وقت باستخدام الأمر (SET ORDER TO) . وذلك حتى يتم تحديث جميع ملفات الفهرس مع أى تعديل لبيانات ملف قاعدة البيانات .

٢١ - ٦ البحث عن سجل معين

يمكن أن نتخيل عملية البحث عن سجل معين فى الملف كأن هناك مؤشرا معينا يتحرك على أرقام السجلات ليقف عند رقم معين . وطالما كان هذا المؤشر موجودا على هذا الرقم ، فان أى عرض للسجلات باستخدام أوامر عرض البيانات المعروفة يؤدى الى عرض بيانات هذا السجل فقط . فمثلا عند كتابة الأمر (GOTO) وبعده رقم السجل المطلوب ، فان المؤشر يتحرك حتى يصل الى هذا السجل . وعند استخدام الأمر (DISPLAY) مثلا ، يلاحظ ظهور بيانات هذا السجل .

ويمكن من خلال البرنامج الوصول الى السجلات التى يطلبها المستخدم . وذلك عن طريق سؤاله عن رقم السجل الـذى يريده ، ثم الذهاب الى هذا السجل وعرض بياناته .

وذلك كالآتى مثلا :

```
GOTO 5
DISPLAY
```

ولكن ماذا لو كان المستخدم غير متذكر لرقم السجل الذى يريده ؟ وهذا هوما يحدث فى الغالب .

في هذه الحالة فان البرنامج يجب أن يتيح للمستخدم الوصول الى السجل أو السجلات المطلوبة عن طريق مطابقة بيانات حقل معين في هذا السجل لقيمة محددة يدخلها المستخدم الى البرنامج . ويقوم البرنامج بعد ذلك بتوجيه المؤشر الى هذا السجل حتى يسهل بعد ذلك عرض بياناته باستخدام أى أمر من أوامر عرض البيانات مثل :

(@ ... SAY, DISPLAY, LIST,?)

وعند استخدام ملف الفهرس فان عملية البحث تكون سريعة خصوصا اذا كان الحقل المطلوب البحث بواسطته هو الحقل الفهرسى (Key Field) .

وهناك عدة أوامر تستخدم في البحث عن السجلات بعضها يشترط فتح ملف الفهرس أولا ، وبعضها الآخر لايشترط ذلك . وسيتم دراستها بالتفصيل فى الأجزاء التالية .

٢١ - ٦ - ١ استخدام الأمر (LOCATE)

يستخدم هذا الأمر في البحث عن سجل معين في قاعدة البيانات سواء كانت مفهرسة (Indexed) ، أو غير مفهرسة (Unindexed) . ويتم ذلك عن طريق اختبار تحقيق هذا السجل لشرط معين . وهذا الشرط يكون عبارة عن علاقة بين البيانات الموجودة فى حقل معين وبين قيمة محددة . وعند تحقيق أى سجل أو مجموعة من السجلات لهذا الشرط ، فان المؤشر يقف عند أول سجل يحقق هذا الشرط . ويمكن التعامل مع البيانات الموجودة فى هذا السجل حسب الحاجة .

وعندما يراد الوصول الى سجل آخر يحقق الشرط يستخدم الأمر (CONTINUE) وهو لايستخدم الا مع الأمر (LOCATE) . فمثلا للوصول الى السجل الخاص برقم الحساب (60789) يستخدم الأمر التالى :

LOCATE FOR Acct_no = '60789'

والأمر (LOCATE) هو أبطأ أوامر البحث ، لأنه لايعتمد على ملف الفهرس (Index File) . ولذلك فهو يبحث فى كل الملف حتى يجد الحقل الذى يحقق الشرط .

٢١ - ٦ - ٢ الأمر (FIND) والأمر (SEEK)

هذان الأمران فى منتهى القوة لأنهما يؤديان الى الوصول الى السجل

المطلوب بسرعة كبيرة جدا مهما كان حجم ملف قاعدة البيانات المستخدم . ولكن استخدامهما يجب أن يتم بمنتهى الحذر . ويتطلب ذلك دراسة كل منهما والتعرف على خصائصه . فمن خصائصهما مثلا أنهما لايعملان الا على ملف مفهرس (Indexed) وأن يكون الحقل الفهرسي (Key Field) هو الحقل الذى يتم البحث عن طريقه .

فمثلا عند البحث عن اسم معين فى قاعدة بيانات الطلبة (Cadets) وليكن (SHEREIF) مثلا يتم كتابة الأمر التالى :

```
FIND SHEREIF
```

وهذا يساوى تماما الأمر

```
SEEK "SHEREIF"
```

ويلاحظ هنا ضرورة وضع السلسلة الحرفية (String) بين علامات تنصيص (Quotation) فى حالة استخدام الأمر (SEEK) فقط . ولكن ليس هناك حاجة لاستخدام علامات التنصيص مع الأمر (FIND) .

ويتم البحث عن الاسم عن طريق مقارنة كل حرف فى هذا الاسم بالحرف المقابل فى بيانات الحقل الفهرسي بدءا من أول حرف والاستمرار حرفا حرفا حتى الوصول الى نهاية الاسم المطلوب البحث عنه . فمثلا عند استخدام الأمر السابق يقف المؤشر عند أول سجل يحتوى حقل الاسم فيه على الاسم (SHEREIF) . فاذا كان هناك حقل يحتوى على الاسم (SHEREIFA) مثلا فان المؤشر يقف عنده، رغم أن المطلوب هو الاسم (SHEREIF) وليس (SHEREIFA) .

واذا أريد الوصول الى السجل المطابق تماما للاسم المطلوب ، يتم استخدام الأمر (SET EXACT ON) كما سيتم الايضاح فيما بعد . ولكن بدون استخدام هذا الأمر ، فان البحث عن سلسلة حرفية معينة أو حرف معين يعنى البحث عن أى حقل يبدأ بهذه السلسلة أو هذا الحرف .

فمثلا للوصول الى أول سجل يبدأ الاسم فيه بالحرف (A) يتم كتابة الأمر التالى :

```
SEEK "A"
```

والأمر (SEEK) أقوى وأكثر شمولا من الأمر (FIND) . حيث أن الأمر (FIND) يتعامل مع المدخلات كحروف فقط . حتى اذا أريد البحث عن عدد معين فانه يتعامل مع هذا العدد كسلسلة حرفية . فمثلا عند كتابة الأمر التالى :

```
FIND 125.60
```

فان البرنامج يقارن جميع مدخلات الحقل الفهرسى حتى يصل الى الرقم المطابق تماما لهذا العدد .

أما الأمـــر (SEEK) فانه يتعامل مع المدخلات الحرفية والعـددية والتاريخية بالاضافة الى أى علاقات (Expressions) . لذلك فمن الضـرورى لادخال قيمة حرفية كتابتها بين علامات تنصيص كما سبق الايضاح .

وللبحث عن تاريخ معين مثلا يتم كتابة الأمر التالى :

```
SEEK CTOD('05/01/90')
```

مع ملاحظة فتح الملف الفهرسي الذى يستخدم حقل التاريخ كحقل فهرسي رئيسى (Master Index) .

واستخدام هذين الأمرين (FIND) و (SEEK) يؤدى الى تحريك المؤشر الى أول ملف قاعدة البيانات بصرف النظر عن مكان هذا المؤشر قبل استخدام الأمر . واذا لم يجد البرنامج القيمة المطابـقة فان المؤشر يتحرك الى نهاية ملف قاعدة البيانات . كما تظهر الرسالة التالية :

```
No Find
```

ويمكن استخدام هـذين الأمرين مع متغيرات الذاكرة ، مع ملاحظة استخـدام التعويض بالماكرو (Macro Substitution) عند استخدام الأمر (FIND) مع متغـيرات الـذاكرة . فمثلا عند ادخال الاسم (Mohamed) فى متغير الذاكرة (mname) كالآتى :

```
INPUT "Mohamed" TO mname
```

فعند استخدام الأمر SEEK فى البحث عن محتويات المتغير mname فى ملف قاعدة البيانات ، يكتب الآتى :

```
SEEK mname
```

أما عند استخدام الأمر (FIND) لتنفيذ نفس الشيء ، فيكتب كالآتى :

```
FIND & mname
```

ويلاحظ هنا استخدام أمر التعويض بالماكرو (&) . وهذا الأمـر سيتم

شرحه بالتفصيل فيما بعد .

والمثال التالى يوضح استخدام الأمر (FIND) من خلال برنامج فرعى (Module).

```
SET TALK OFF
CLEAR
m_name = SPACE(20)
@ 5,5 SAY 'Enter a name'
@ 5,30 GET m_name PICTURE "@A"
READ
STORE LTRIM(TRIM(m_name)) TO m_name
USE Cadets INDEX name
FIND & m_name
CLEAR
@ 1,5 SAY TRIM(name)
@ 3,5 SAY address
RELEASE m_name
RETURN
```

والسطر الأول من البرنامج يؤدى الى عدم ظهور خطوات تنفيذ البرنامج على الشاشة كما سبق الايضاح .

والسطر الثانى يؤدى الى مسح الشاشة .

والسطر الثالث يؤدى الى انشاء متغير ذاكرة (m_name) طوله عشرون حرفا .

والسطر الرابع يؤدى الى عرض رسالة للمستخدم لتوضيح المطلوب .

والسطر الخامس يؤدى الى ظهور عمود ضوئى بطول عشرين حرفا حتى يقوم المستخدم بادخال الاسم المطلوب فيه .

والسطر السادس يؤدى الى تخزين الاسم الذى يدخله المستخدم فى متغير الذاكرة (m_name) .

والسطر السابع يؤدى الى التخلص من المسافات الموجودة فى اول الاسم او فى آخره .

والسطر الثامن يؤدى الى فتح ملف قاعدة البيانات (Cadets) وملف الفهرس المرتبط به (name) .

والسطر التاسع يؤدى الى البحث عن الاسم الذى يدخله المستخدم

ويلاحظ هنا استخدام أمر التعويض بالماكرو (Macro Substitution) .

والسطر العاشر يؤدي الى مسح الشاشة تمهيدا لعرض البيانات الخاصة بالسجل الذى يتم الوصول اليه .

والسطران الحادى عشر والثانى عشر يؤديان الى عرض الاسم والعنوان الخاص بهذا السجل .

والسطر الثالث عشر يؤدى الى الغاء متغير الذاكرة (m_name) حتى يستطيع المستخدم ادخال اسم آخر للبحث عنه .

والسطر الأخير يؤدي الى العودة الى البرنامج الرئيسي الذى قام باستدعاء هذا البرنامج الفرعى .

ويجب ملاحظة أن هذا البرنامج يصل فقط الى أول سجل يحتوى على الاسم الذى يكتبه المستخدم . وبمجرد وقوف المؤشر عند هذا السجل ، يتم عرض الحقول المطلوبة من هذا السجل باستخدام الأمر (@...SAY) كما يتضح من البرنامج .

٢١-٦-٢ عرض بيانات جميع السجلات التى تحقق الشرط

كما سبق الايضاح فى البرنامج السابق . فان الأمرين (SEEK) ، (FIND) يؤديان الى وضع المؤشر عند أول سجل يحقق الشرط . وبالتالى يمكن عرض بيانات هذا السجل .

ولكن ماذا لو أراد المستخدم عرض جميع السجلات التى تحقق الشرط ؟ . وذلك كما يحدث مع الأمر (LOCATE) باستخدام الأمر (CONTINUE) . في هذه الحالة يلزم استخدام وسيلة معينة لتحريك المؤشر من أول سجل يحقق الشرط الى السجل الثاني والثالث و ... وهكذا . ويتم ذلك عن طريق استخدام الحلقة التكرارية باستخدام الأمر (DO WHILE) .

فمثلا البرنامج التالى تعديل للبرنامج السابق بحيث يحقق المطلوب :

```
SET TALK OFF
CLEAR
m_name = SPACE(20)
@ 5,5 SAY 'Enter a name'
```

```
@ 5,30 GET m_name PICTURE "@A"
READ
STORE LTRIM(TRIM(m_name)) TO m_name
USE Cadets INDEX name
FIND & m_name
CLEAR
r = 1
DO WHILE TRIM(name) = m_name
    @ r,10 SAY TRIM(name)
    @ r+1,10 SAY address
    ?
    SKIP
    r = r+4
ENDDO WHILE TRIM(name) = m_name
RELEASE m_name
RETURN
```

ويلاحظ أن الجزء الأول من البرنامج مطابق للبرنامج السابق تماما . ثم تمت اضافة الحلقة التكرارية التي تبدأ بالأمر (DO WHILE) . وقد تم انشاء متغير ذاكرة (r) قبل بداية الحلقة التكرارية لاستخدامه في تحديد السطر الذى يتم استخدامه فى عرض البيانات على الشاشة .

والجزء الأول من البرنامج يؤدى الى الوصول الى أول سجل يحتوى على الاسم الذى يكتبه المستخدم كما سبق الايضاح من البرنامج السابق .

أما الجزء الخاص بالحلقة التكرارية فانه يؤدي الى تحريك المؤشر الى باقى السجلات التى تحقق الشرط وعرض بياناتها واحدا تلو الآخر .

ويتم ذلك عن طريق استخدام هذا الشرط في أول الحلقة التكرارية . أى أن الحلقة التكرارية تبدأ فى عرض بيانات الاسم والعنوان للسجل الثانى الذى يحقق الشرط . ثم عن طريق الأمر (SKIP) يتم نقل المؤشر الى السجل التالى ثم تتكرر الحلقة التكرارية . وفى كل مرة يتم اختبار الشرط فى أول الحلقة . وبالتالى لايتم عرض بيانات الاسم والعنوان الا للسجلات التى تحقق الشرط .

ويلاحظ هنا استخدام الأمر :

```
r = r + 4
```

وذلك لزيادة عدد السطور ، وبالتالى ظهور بيانات السجلات وبينها سطور خالية . ويجب ملاحظة أن أى كلمات مكتوبة بعد الأمر

(ENDDO) هي ملاحظات تظهر فقط لمخطط البرامج عند عرض الأوامر على الشاشة ولكنها لاتؤثر على تشغيل البرنامج . وهي تفيد في توضيح أي أمر (DO WHILE) يتبع له هذا الأمر (ENDDO) . وهذا يكون ضروريا بصفة خاصة في البرامج التي تحتوى على عدة حلقات تكرارية حتى يكون التسلسل المنطقى للبرنامج واضحا .

وهذا البرنامج يؤدى الى عرض جميع السجلات التي تحقق الشرط على الشاشة .

ملاحظة

عندما يزيد عدد السجلات عن عدد سطور الشاشة ، أي يصبح (r) أكبر من (٢٤) . في هذه الحالة يتوقف ظهور باقي السجلات ، وتظهر رسالة خطأ . لذلك يلزم اضافة أوامر أخرى للبرنامج تؤدى الى ظهور شاشة جديدة عند الوصول الى آخر سطر في الشاشة . وهذا سيتم شرحه فيما بعد .

٢١ - ٧ اختبار نهاية الملف

عند استخدام أي أمر من أوامر البحث مثل (SEEK) أو (FIND) أو (LOCATE) فمن المهم اختبار نهاية الملف (End of File) . وذلك لأن البرنامج عند بحثه عن قيمة معينة سواء كانت حرفية أو عددية فانه يبدأ البحث من أول سجل في الملف ويستمر في فحص السجلات واحدا بعد الآخر . فاذا لم يجد أى سجل يحقق الشرط ، فان المؤشر يصل الى نهاية الملف . وقد يسبب هذا الحصول على نتائج غير سليمة خصوصا عند استخدام بعض الأوامر التي تبدأ البحث من المكان الذي يقف عنده المؤشر . لذلك فان برنامج (DBase III+) يتيح لمخطط البرامج اختبار نهاية الملف عن طريق دالة خاصة تسمى EOF() . هذه الدالة تعطى القيمة صحيح (.T.) أي (True) أو غير صحيح (.F.) أي (False) ، بناء على وضع المؤشر اذا كان في نهاية الملف أو ليس في نهاية الملف .

ملاحظة

هناك فرق بين نهاية الملف (End of File) وبين آخر سجل في قاعدة البيانات (BOTTOM) . حيث أن نهاية الملف تكون نقطة بعد آخر سجل مباشرة . وكذلك فان بداية الملف تختلف عن أول سجل في الملف (TOP) .

وعند كتابة برنامج مثل البرنامج السابق ، يجب اضافة مجموعة من الأوامر التي

تؤدى الى اختبار الوصول الى نهاية الملف .

ولتوضيح ذلك يمكن كتابة نفس البرنامج السابق كالآتى :

```
SET TALK OFF
m_name = SPACE(20)
@ 5,5 SAY 'Enter a name'
@ 5,30 GET m_name PICTURE "@A"
READ
STORE LTRIM(TRM(m_name) TO m_name
USE Cadets INDEX name
FIND & m_name
      IF EOF()
            CLEAR
            @ 5,1 SAY 'Sorry, there is no' + m_name
            @ 15,1 WAIT
      ELSE
            CLEAR
            r = 1
            DO WHILE TRIM (name) = m_name.AND..NOT.EOF()
                @ r,10 SAY TRIM(name)
                @ r+1,10 SAY address
                ?
                SKIP
                r = r + 4
            ENDDO WHILE TRIM(name) = m_name
       ENDIF
RELEASE m_name
RETURN
```

ويلاحظ هنا أن هناك نقطتين يتم من خلالهما اختبار نهاية الملف . النقطة الأولى فى السطر التالى للأمر (FIND) مباشرة باستخدام الأمر (IF) . والنقطة الثانية فى بداية الحلقة التكرارية عندما ينتقل المؤشر الى سجل جديد باستخدام الأمر (SKIP) . حيث يتم اختبار نهاية الملف عند الرجوع الى أول الحلقة التكرارية (DO WHILE).

٢١ - ٨ استخدام دالة رقم السجل

تستخدم الداله (RECNO()) لتحديد رقم السجل الذى يقف عنده المؤشر . ويمكن تخزين هذا الرقم فى متغير ذاكرة واستخدامه فيما بعد وذلك كالآتى مثلا :

```
FIND & m_name
IF .NOT. EOF()
     STORE RECNO() TO record_n
ENDIF
```

فى هذه الحالة يتم البحث عن محتويات المتغير (m_name) فى ملف قاعدة البيانات . فاذا وجد البرنامج سجلا يحتوى على هذه المحتويات ، فانه يخزن رقم هذا السجل فى متغير الذاكرة (record_n) ، واذا لم يجده يصل الى نهاية الملف .

ويمكن بعد ذلك الذهاب الى السجل الذى تم تخزين رقمه فى المثال السابق باستخدام الأمر (GOTO) كالآتى :

```
GOTO record_n
```

كما يمكن عرض بيانات هذا السجل أو مسحه حسب الحاجة .

٢١ - ٩ استخدام الدالة (FOUND())

فى بعض الأحيان يكون مطلوبا فقط مجرد معرفة اذا كان السجل المطابق موجودا أم لا . وليس مهما الوصول الى هذا السجل أو عرض بياناته . فمثلا عندما يقوم المستخدم بادخال حساب معين ، فان البرنامج يجب أن يتأكد أن هذا الرقم لم يسبق ادخاله حتى لايتم ادخال سجلات مكررة .وفى هذه الحالة يمكن كتابة السطور التالية :

```
SEEK macct
IF FOUND()
     @ 10,10 SAY "This number already exists"
ELSE
     EXIT
ENDIF
```

والداله (FOUND()) فى هذا المثال تعطى القيمة صحيح (True) أو غير صحيح (False) . فاذا أعطت القيمة صحيح فان ذلك يعنى أن الأمر (SEEK) قد وجد رقم حساب يطابق الرقم الذى تم ادخاله فى المتغير (macct) . وفى هذه الحالة تظهر الرسالة المبينة للمستخدم لتحذره من ادخال هذا الرقم .

ويلاحظ هنا أنه ليس مهما تحديد السجل المطابق للرقم ، ولكن المهم معرفة اذا كان هناك سجل مطابق أم لا .

٢١ - ١٠ استخدام المرشح (Filter)

يمكن استخدام المرشحات (Filters) كوسيلة أخرى لتجميع السجلات التي تحقق شرطا معينا . وذلك عندما يراد اجراء عملية معينة على هذه السجلات مثلا مثل تجميع الأعداد الموجودة في حقل معين . ويستخدم لانشاء هذه المرشحات الأمر (SET FILTER TO) ثم كتابة الشرط أو الشروط المطلوب استخدامها وذلك كالآتي مثلا :

```
SET FILTER TO M $ name .AND. age < 30
```

والشرط الأول (M $ name) يعني كل الأسماء التي تحتوى على الحرف (M) .

والشرط الثاني يعني الأشخاص الذين تقل أعمارهم عن ٣٠ سنة . وهذا السطر يؤدى الى تصفية قاعدة البيانات وعدم السماح بالمرور من المرشح الا للأسماء التي تحقق الشرطين معا . أي أسماء الأشخاص الذين تحتوى أسماؤهم على الحرف (M) وتقل أعمارهم عن ٣٠ سنة .

ملاحظة

يراعى نقل المؤشر الى أول الملف بعد استخدام الأمر (SET FILTER TO) وقبل اجراء أى عملية أخرى . وذلك لأن هذا الأمر مثل الأمر (SEEK) أو الأمر (FIND) يقوم بالمرور على جميع السجلات . وبالتالى فانه ينقل المؤشر الى آخر الملف . ولذلك فعند اجراء أى عملية مثل تجميع حقل عددي مثلا ، فان التجميع يبدأ من مكان المؤشر وبالتالى لاتكون النتيجة سليمة . ولذلك يستخدم الأمر (GO TOP) بعد استخدام المرشح مباشرة للوصول الى بداية الملف مرة أخرى .

٢١ - ١١ استخدام الدالة (DELETED())

عندما يراد مسح مجموعة من السجلات، فان برنامج (DBase III+) يتيح لمخطط البرامج مسح السجلات على مرحلتين . المرحلة الأولى يتم خلالها وضع علامات على السجلات المطلوب مسحها . والمرحلة الثانية يتم خلالها تنفيذ عملية المسح . ويتيح ذلك للمستخدم التاكد أنه يريد مسح هذه السجلات قبل المسح الفعلى لها .

كما يتيح له أيضا الاحتفاظ بالسجلات المطلوب مسحها أطول فترة ممكنة قبل مسحها فعليا من قاعدة البيانات .

وفى هذه الحالة يجب عزل هذه السجلات حتى لاتؤثر فى سرعة البحث عن أى

سجل . وحتي يمكن اجراء أى عمليات على السجلات المطلوبة فقط دون اضاعة الوقت فى تنفيذها على سجلات مطلوب مسحها .

ويتم عزل السجلات المطلوب مسحها بطريقتين :

الطريقة الأولى باستخدام الأمر (SET DELETED ON) . حيث أن الوضع المبدئي (Default) لهذا الأمر يكون (OFF). والطريقة الثانية باستخدام الدالة DELETED() كالآتى :

```
SET FILTER TO .NOT. DELETED()
```

وهذا يؤدى الى عزل السجلات التى سبق تجهيزها للمسح .

٢١ - ١٢ استخدام الأمر (SET EXACT ON)

عندما يقارن برنامج (DBase III+) بين سلسلتين حرفيتين (Strings) ، فانه يقارن كل حرف من السلسلة الأولى بالحرف المقابل له من السلسلة الثانية . وتستمر هذه المقارنة حتى تنتهى السلسلة اليمنى . فمثلا عند مقارنة متغير يحتوى على الاسم "Mohamed" بمتغير آخر يحتوى على الحروف "Moh" يتم كتابة السطر التالى :

```
? "Mohamed" = "Moh"
```

وعند الضغط على مفتاح الادخال تظهر القيمة (.T.) أى أن المقارنة صحيحة . وذلك بالرغم من عدم تطابق الطرفين .

وذلك لأن البرنامج يقارن أولا الحرف الأول فى الطرفين فيجده مطابقا، فينتقل الى الحرف الثانى فيجده مطابقا ثم ينتقل الى الحرف الثالث فيجده مطابقا . ثم لايجد حروفا أخرى فى الطرف الأيمن لذلك يعطى القيمة (.T.) ، أى صحيح .

واذا استبدلنا الطرفين فى السطر السابق ، أى تمت كتابته كالآتى مثلا :

```
? "Moh" = "Mohamed"
```

فان النتيجة فى هذه الحالة تصبح العكس . وذلك لأن الطرف الأيسر ينتهى قبل الأيمن . ولذلك يعطى البرنامج القيمة (.F.) أى غير صحيح (False) .

وعند استخدام الأمر (SEEK) أو الأمر (FIND) يحدث نفس الشئ . فمثلا عند البحث عن الحروف (Moh) فى حقل الاسم يستخدم الأمر التالى :

```
SEEK "Moh"
```

وفي هذه الحالة يقارن البرنامج بين كل اسم موجود في الحقل وبين الحروف "Moh". ثم يقف عند أول اسم يبدأ بهذه الحروف . وهذا قد يكون مطلوبا في بعض الأحوال ، عندما يراد مثلا البحث عن الاسم الذى يبدأ بهذه الحروف . ولكن في أحوال أخرى قد يكون مطلوبا البحث عن اسم محدد . وفى هذه الحالة يستخدم الأمـــر (SET EXACT ON) . وهذا الأمر يساعد على البحث عن سلسلة حرفية معينة حتى يصل البرنامج الى سلسلة حرفية مطابقة لها تماما فى الملف .

ملاحظة

يراعى عند استخدام هذا الأمر (SET EXACT ON) أن يتم استخدامه فى البحث المطلوب ثم اعادته الى الوضع المبدئى (Default) مرة أخرى بمجرد الانتهاء من البحث . وذلك عن طريق الأمر (SET EXACT OFF) .

٢١ - ١٣ منع الازدواج (Duplication)

فى بعض قواعد البيانات تكون هناك بعض الحقول المنفردة (Unique) أى التى تحتوى على قيمة مختلفة لكل سجل من سجلات الملف . فمثلا فى قواعد البيانات الخاصة بالحسابات يكون هناك رقم حساب (.Account no) مستقل لكل سجل . وفى العادة يتم فهرسة الملف بناء على هذا الحقل المنفرد (Unique) . ولكن عند ادخال البيانات فقد يقوم المستخدم بادخال بيانات نفس السجل مرتين . لذلك يستخدم الأمر (SET UNIQUE ON) . فعندما يستخدم هذا الأمر قبل فهرسة الملف (Indexing) فان برنامج (+ DBase III) يقوم بادخال السجلات المنفردة (Unique) فقط فى الفهرس ، ويستبعد أى سجلات مكررة . فعند البحث عن سجلات لعرضها أو تعديلها فان المستخدم لن يرى أى سجلات مكررة . واذا كانت هناك سجلات مكررة فعلا ، فان الفهرس لن يظهرها لأنها ليست جزءا من الملف الفهرسى .

كما يمكن تنفيذ هذه العملية عند انشاء ملف الفهرس كالآتى :

```
INDEX ON name TO Name Unique
```

حيث يؤدى هذا الأمر الى انشاء فهرس منفرد (Unique) .

ويجب ملاحظة اعادة الوضع المبدئى (Default) عن طريق كتابة الأمـــر (SET UNIQUE OFF) . وذلك عندما يراد استخدام كل السجلات سواء كانت مكررة أو غير مكررة .

ملاحظة

ما سبق ذكره في هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، FoxBase + ، FoxPro .

الباب الثاني والعشرون

التعامل مع البيانات

عند تعديل بيانات سجل معين أو اضافة سجل جديد ، فان من المهم اعطاء المستخدم الفرصة لمراجعة البيانات التى أدخلها وذلك بعرضها على الشاشة . ثم اعطائه الاختيار بين تخزين البيانات التي كتبها اذا كانت صحيحة ، أو تصحيحها مرة ثانية اذا وجد بعض الأخطاء . ويتم تنفيذ ذلك باستخدام متغيرات الذاكرة (Memory Variables). وتستخدم هذه المتغيرات كحلقة اتصال بين المستخدم وملف قاعدة البيانات ، حيث يتم ادخال البيانات التى يكتبها المستخدم أولا فى متغيرات الذاكرة . ويقوم البرنامج بتحديد السجل المطلوب تعديله ثم استبدال محتويات كل حقل من حقول هذا السجل بمحتويات متغير الذاكرة المقابل له .

ويمكن توضيح هذه العملية فى خطوات محددة كالآتى :

١ - يتم أولا انشاء متغيرات الذاكرة (Memory Variables) لاستقبال مدخلات المستخدم . وكل متغير من هذه المتغيرات يقابل حقلا معينا من حقول قاعدة البيانات . وفى معظم البرامج يتم اعطاء هذه المتغيرات نفس اسم الحقل مع اضافة الحرف m مثل mname أو m_name .
٢ - يتم فتح ملف قاعدة البيانات وملفات الفهرس المرتبطة به . ثم يتم استخدام أى أمر من أوامر البحث للوصول الى السجل المطلوب تعديله مثل (GOTO) ، (LOCATE) ، (FIND) ، (SEEK) .
٣ - يتم عرض شاشة الادخال التى سبق تصميمها . ويلاحظ ظهور محتويات الحقول الخاصة بهذا السجل فيقوم المستخدم بتعديل بيانات هذه الحقول .
٤ - فى حالة اضافة سجل جديد يتم عرض شاشة خالية باستخدام الأمر (APPEND BLANK).
٥ - بعد أن يضيف المستخدم البيانات المطلوبة ، يتم عرض رسالة (Prompt) لسؤال المستخدم عن صحة البيانات التى أدخلها . واذا لم تكن صحيحة يتم اعطاؤه الفرصة لتعديل هذه البيانات وتصحيح الأخطاء .
٦ - يتم استبدال البيانات الموجودة فى الحقول بالبيانات الموجودة في متغيرات الذاكرة كل حسب الحقل المقابل له . وذلك باستخدام الأمر (REPLACE) .
٧ - يتم اغلاق الملف ومسح متغيرات الذاكرة حتى يمكن استخدامها مرة ثانية .

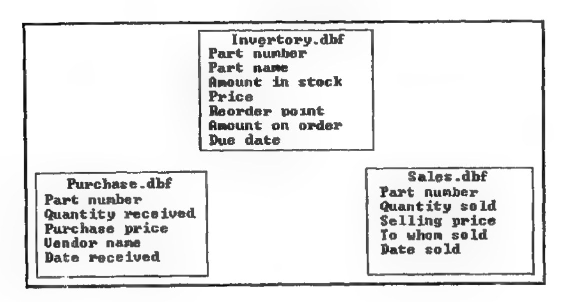
٣٢ - ١ التعديل المجمع (Batch Updating)

فى كثير من نظم المعلومات ، يستخدم النظام الطريقة التجميعية فى تعديل البيانات (Batch Updating) . ويظهر ذلك بوضوح فى معظم نظم الحسابات (Accounts) والبنوك والمخازن . حيث يتم تجميع التعديلات فى ملف منفصل يسمى ملف الحركة (Transaction File) ، ثم يتم ادخال هذه التعديلات على الملف الرئيسي (Master File) كل فترة . وقد تكون هذه الفترة يوما أو أسبوعا أو شهرا على حسب حجم التعديلات التى يتم ادخالها .

ولنأخذ مثلا نظام المخازن (Inventory) كمثال لهذه النظم . حيث يتكون

نظام المخـــازن البسيط من ملف رئيسي (Master File) وملفـين للحــــركة (Transaction Files).

انظر الشكل (٢٢ - ١)



شكل (٢٢ - ١)نظام المخازن

ويلاحظ من الشكل تركيب الملف الرئيسي (Inventory.dbf) وكذلك تركيب ملفات الحركة (Sales.dbf) ، (Purchase.dbf) . والملف الرئيسي هنا يحتوى على البيانات الحالية للبضائع المخزنة (Stock) . وهو يستمد معلوماته من بيانات حركة البيع (Sales Transactions) التي تصل اليه من ملف المبيعات (Sales.dbf) وكـــذلك مــــن بيانـــــات حركة الشراء (Purchase transactions) التـي تصل اليه من ملف المشتريات (Purchase.dbf).

ويلاحظ من الرسم أيضا وجود حقل مشترك بين الملفات الثلاثة ، وهو حقل رقم الجزء (Part number).هذا الحقل هو الذى يتم عن طريقه ربط الملفات الثلاثة .

ولكي يتم تعديل الملف الرئيسي من ملف المبيعات مثلا يتم أولا فهرسة الملفات باستخدام حقل رقم الجزء كحقل فهرسي (Index Key) وذلك كالآتي :

```
USE Inventory
INDEX ON part_no TO Master
USE Sales
INDEX ON part_no TO Sales
```

ثم يتم فتح كل ملف في منطقة عمل مختلفة باستخدام الأمر (SELECT) كما سبق الايضاح . ويتم استخدام الأمر (UPDATE) فى تعديل حقل الكمية الموجودة مثلا (on_hand) ، وذلك كالآتى :

```
CLEAR ALL
SELECT 2
USE Sales INDEX Sales
SELECT 1
USE Inventory INDEX Master
UPDATE ON part_no FROM Sales REPLACE ;
on_hand WITH on_hand - B -> Qty
```

ملاحظة

عند زيادة طول السطر عن عرض الشاشة أثناء كتابة البرنامج ، يتم اضافة الحرف (;) في نهاية السطر . وهذا الحرف يخبر البرنامج أن المكتوب فى السطر التالى تكملة لهذا السطر . ويلاحظ ذلك فى السطر السادس من البرنامج .

ويلاحظ من البرنامج فتح كل ملف في منطقة عمل (Work Area) مختلفة . كما يلاحظ أن آخر منطقة عمل يتم اختيارها هي التى يتم العمل فيها . وهذا يعنى أنــه عند العمل فى أى منطقة يجب أولا اختيار هذه النطقة باستخدام الأمر (SELECT) .

ويلاحظ أيضا استخدام الأمر (UPDATE) في تعديل الكمية الموجودة في الحقل (on_hand) في الملف الرئيسى (Inventory.dbf) عن طريق طرح الكمية الموجودة فى الحقل (Qty) الخاص بملف المبيعات (Sales.dbf) من الكمية الموجودة في الحقل (on_hand) في الملف الرئيسى .

ويجب ملاحظة أن الحرف (B) هنا يمثل منطقة العمل (2) التى تحتوى على ملف المبيعات . كما أن العلامة (<-) تعنى الحصول على محتويات الحقل (Qty) من ملف المبيعات الموجود في منطقة العمل (B) .

ويمكن استخدام نفس الطريقة فى تعديل الملف الرئيسى بناء على بيانات ملف المشتريات (Purchase.dbf) وذلك بكتابة الأمر التالى :

```
UPDATE ON part_no FROM Purchase REPLACE on_hand WITH ;
on_hand + B -> Qty
```

ويلاحظ هنا اضافة الكمية التى تم شراؤها الى الكمية الفعلية الموجودة فى المخزن .

ملاحظة

يتم فى كتاب التطبيقات (الكتاب الثالث فى المجموعة) شرح برنامج مخازن كامل .

٢٢ - ٢ مسح السجلات

يتم مسح السجلات المطلوب مسحها باستخدام الأمر (DELETE) . وتتبع فى ذلك نفس الخطوات التى سبق ايضاحها عند تعديل السجلات . حيث يتم أولا البحث عن السجل المطلوب باستخدام أوامر البحث السابق شرحها . أى يتم وضع المؤشر على السجل المطلوب مسحه ثم كتابة الأمر (DELETE) . أو يتم تنفيذ ذلك بكتابة سطر واحد كالآتى :

```
DELETE ALL FOR YEAR(Date) = myear
```

حيث (myear) هو متغير ذاكرة يتم تخزين الرقم الممثل لسنة معينة فيه .

كما أن القيمه YEAR(Date) هى القيمة العددية الممثلة للسنة المخزنة فى الحقل (Date).

وكلمة (ALL) تحدد المدى (Scope) المطلوب البحث خلاله .

ويؤدى هذا الى مسح جميع السجلات التى تختص بسنة معينة يقوم المستخدم بتحديدها . ويفيد ذلك عندما يراد مثلا مسح بيانات سنين سابقة بعد انتهاء الحاجة اليها .

ويجب ملاحظة أن الأمر (DELETE) وحده لايؤدى الى مسح السجلات مباشرة . ولكنه يميز هذه السجلات تمهيدا لمسحها باستخدام الأمر (PACK) فى أى وقت .

ملاحظة

بعض مخططى البرامج يفضلون عدم استخدام الأمر (PACK) يوميا . ويقومون بتجميع السجلات المطلوب مسحها كل فترة ثم مسحها مرة واحدة عند التأكد من عدم الحاجة اليها .

وعندما يراد مسح بيانات الملف بالكامل يستخدم الأمر (ZAP) . ويجب الحرص

عند استخدام هذا الأمر لأنه يمسح السجلات مباشرة دون الحاجة الى استخدام الأمر (PACK) . فمثلا عندما يراد مسح البيانات التى سبق ادخالها في ملفات المخازن والبدء في تسجيل بيانات جديدة ، تستخدم الأوامر التالية :

```
USE Inventory
ZAP
USE Sales
ZAP
USE Purchase
ZAP
CLOSE DATABASES
```

ملاحظة

مسح بيانات الملف لايعني مسح الملف بالكامل . حيث أن هيكل الملف (File Structure) يظل كما هو .

٢٢ - ٣ نسخ السجلات

قبل مسح بعض السجلات التى انتهت الحاجة اليها ، قد يكون من المفيد نسخ هذه السجلات في ملف أرشيف (Archive File) . فمثلا في برنامج المخازن قد تكون البيانات التي مضى عليها أكثر من خمس سنوات غير مطلوبة ويراد مسحها . ولكن يراد الاحتفاظ بها في ملف أرشيف حتى يمكن الرجوع اليها وقت الحاجة . ويستخدم الأمر (COPY) في نسخ سجلات من ملف الى ملف آخر . فمثلا في برنامج المخازن السابق يمكن كتابة الأوامر التالية :

```
USE Inventory
COPY TO file1 FOR YEAR(Date) = myear
DELETE ALL FOR YEAR(Date) = myear
PACK
```

وتؤدى هذه الأوامر الى نسخ جميع السجلات التى تختص بسنة معينة الى ملف الأرشيف (File1) . ثم يتم بعد ذلك مسح هذه السجلات نفسها من ملف قاعدة البيانات الأصلى .

٢٢ - ٤ التعامل مع الملفات المرتبطة

كما سبق الايضاح ، فان الملفات المرتبطة (Related) هى ملفات يربط بينها

حقل مشترك . وهي تساعد على تقسيم ملف قاعدة البيانات الكبير الى عدة ملفات صغيرة نسبيا لتقليل التحميل (Overload)على الذاكرة المؤقتة (RAM) وبالتالي زيادة سرعة التشغيل .

وعندما يراد الحصول على بيانات من عدة ملفات مرتبطة يجب أولا ربط هذه الملفات ببعضها . وذلك لأن المؤشر الخاص بكل ملف يكون موضوعا على سجلات مختلفة . وبالتالى لايمكن عرض بيانات سجل معين بتجميعها من عدة ملفات الا بعد وضع المؤشر في جميع هذه الملفات عند نفس السجل .

ولتوضيح ذلك نفرض أن هناك ملفين اسمهما (First.dbf) ، (Second.dbf). ونفرض أن الملف (Second.dbf) يحتوى على الحقول (name, birth_d) والملف (First.dbf) يحتوى على الحقل (address) . ويراد عرض بيانات هذه الحقول . فيتم كتابة الأوامر التالية :

```
SELECT 1
USE First INDEX First
SELECT 2
USE Second INDEX Second
DISPLAY name, birth_d, A -> address
```

حيث A هو الحرف المرادف (ALIAS) للمنطقة الأولى (١) .

فى هذه الحالة يتم عرض بيانات حقول الاسم (name) وتاريخ الميلاد (birth_d) من الملف (Second.dbf) ، كما يتم عرض العنوان (address) من الملف (First.dbf) .

ولكن هل هذه البيانات تختص بنفس السجل ؟ والاجابة على هذا السؤال بالنفي . وذلك لأنه لم يتم ربط المؤشر الخاص بالملف (First.dbf) بالمؤشر الخاص بالملف (Second.dbf) . وبالتالى ليس هناك ما يضمن وقوف المؤشر على نفس السجل فى الملفين .

ولكى يتم ربط الملفات ببعضها يستخدم الأمر (SET RELATION) .

٢٢ - ٥ استخدام الأمر (SET RELATION)

هذا الأمر يربط بين ملفين عن طريق حقل مشترك فيهما . ويؤدى هذا الى ربط

المؤشر الخاص بالملف الثاني بالمؤشر الخاص بالملف الاول . وبعد تحريك المؤشر في الملف الأول للوصول الى سجل معين باستخدام أوامر البحث السابق شرحها، فان المؤشر الخاص بالملف الثاني يتحرك معه تماما ، وبالتالي فعند اجراء أي عملية على بيانات هذا السجل يمكن الحصول على بياناته من الملفين في نفس الوقت .

فمثلا في الملفين (First.dbf) ، (Second.dbf) نفرض أنه يوجد حقل مشترك وهو حقل (emp_no) أى رقم الموظف . فان الربط بين الملفين يتم عن طريق كتابة الأوامر التالية :

```
SELECT 1
USE First INDEX First
SELECT 2
USE Second INDEX Second
SET RELATION TO emp_no INTO First
DISPLAY name, birth_d, A -> address
```

1	2	3
First.dbf Name Birth-d Emp-no	Second.dbf Address Emp-n	

شكل (٢٢ - ٢)

ويجب ملاحظة أنه لايمكن استخدام أكثر من علاقة واحدة في كل منطقة عمل (Work area) . كما أن هذه العلاقة تنتهي بمجرد اغلاق منطقة العمل أو استخدام الأمر (SET RELATION TO) دون كتابة أي شيء بعده . كما يجب ملاحظة أن الملف الذي يتم ربطه ، والذي يكتب اسمه بعد (INTO) يجب أن يكون مفهرسا (Indexed) على الحقل المشترك بين الملفين . أما الملف الآخر فلا يشترط أن يكون مفهرسا .

واذا رجعنا الى البرنامج السابق فاننا نلاحظ أن منطقة العمل (2) هي منطقة العمل الفعالة (Active) التي يجرى العمل عليها ، وذلك لأنها آخر منطقة عمل تم اختيارها بواسطة الأمر (SELECT) . لذلك فان أي حركة لمؤشر الملف في هذه المنطقة يتبعها نفس الحركة لمؤشر الملف في المنطقة الأولى . وبالتالي يمكن الحصول على بيانات لنفس السجل من الملفين .

ملاحظة

في حالة انشاء علاقة بين ملفين يجب أن يكون الحقل المشترك منفردا (Unique) أي غير متكرر .

٢٢ - ٦ استخدام ملف المنظر (View File)

يستخدم ملف المنظر (View File) في ربط الملفات كما سبق الايضاح في الجزء الخاص ببرنامج المساعد (Assistant) . وهو يساعد على تخزين العلاقة بين ملفين في ملف منفصل .وهذا عكس استخدام الأمر (SET RELATION TO) الذى يؤدى الى ربط الملفين ربطا مؤقتا .

ويمكن انشاء عدة ملفات منظر كل منها يحتوى على مجموعة من الحقول المطلوب عرضها على الشاشة أو طباعتها على الطابعة . وبمعنى آخر يمكن النظر الى قاعدة البيانات من زوايا مختلفة (Views) .

ويمكن انشاء ملفات المنظر من برنامج المساعد (Assistant) كما سبق الايضاح أو بكتابة الأمر (CREATE VIEW) كما يمكن تعديله أو انشاؤه بواسطة الأمر (MODIFY VIEW) . وفي هذه الحالة تظهر نفس الشاشات التى سبق ايضاحها فى استخدام برنامج المساعد (Assistant) .

كما يمكن انشاء ملف المنظر من خلال البرنامج عن طريق كتابة الأمر (CREATE VIEW FROM INVIRONMENT) . وذلك بعد ادخال العلاقة المطلوبة عن طريق الأمر (SET RELATION TO) .

وفي جميع الأحوال يتم استخدام هذا الملف عن طريق كتابة الأمر (SET VIEW TO) ثم كتابة اسم الملف . وفى المثال السابق مثلا يمكن كتابة الآتى :

```
SET RELATION TO emp_no INTO first
CREATE VIEW F_second FROM INVIRONMENT
```

ويؤدى هذا الى انشاء ملف المنظر (F_second) . هذا الملف يحتوى على الآتى :

١ - كل ملفات قواعد البيانات وملفات الفهرس ورقم كل منطقة عمل (Work Area) .
٢ - العلاقة أو العلاقات الموجودة بين الملفات .
٣ - ملف التشكيل (Format File) المفتوح .

ويمكن التحكم في أسماء الحقول التي تظهـر في هـذا الملـف باستخـدام الأمـر (SET FIELDS TO) ثم كتابة أسماء الحقول المطلوب عرض بياناتها .

وهذا الأمر يؤدي الى تصفية حقول الملفات الموجودة في ملف المنظر بحيث لاتظهر الا بيانات الحقول التي يتم تحديدها .

ويجب ملاحظة استخدام الأمر (SET FIELDS ON) عندما يراد استخدام هذه الحقول التي تم اختيارها . ويمكن اضافة ذلك الى المثال السابق فيصبح كالآتي :

```
SET RELATION TO emp_no INTO first
SET FIELDS TO name, birth_d, A -> address
CREATE VIEW F_second FROM INVIRONMENT
SET FIELDS ON
DO WHILE .NOT. EOF()
     DISPLAY
     SKIP
ENDDO
```

ويمكن الغاء عملية تصفية الحقـول باستخدام الأمر (SET FIELDS OFF) كما يمكـن تنفيـذ ذلـك أيضـا باسـتخـدام الأمر (CLEAR FIELDS) أو (SET FIELDS TO) دون كتابة أي شيء بعده .

مـلاحظة

ما سبق ذكره في هذا الباب ينـطبق أيضا على كل برامج عائلة (DBase) مثل (DBase IV) ، (FoxBase +) ، (FoxPro) .

الباب الثالث والعشرون

الطبـــاعـة

هناك أوامر تساعد على توجيه مخرجات البرنامج الى الشاشة أو الطابعة حسب الحاجة . ولكن فى جميع الأحوال يجب التأكد أولا أن الطابعة قد تم تشغيلها وتوصيلها بالجهاز .

٢٣ - ١ أوامر الطباعة

هناك أمران يستخدمان فى توجيه البيانات الى الطابعة . ورغم أنهما يحققان نفس الهدف ، الا أنهما يختلفان فى خصائصهما . ولذلك يجب الالمام بخصائص كل منهما حتى يمكن معرفة أى الأمرين يجب استخدامه وفى أى مكان من البرنامج .

وهذان الأمران هما (SET DEVICE TO PRINT) و (SET PRINT ON)

٢٣ - ٢ استخدام الأمر (SET DEVICE TO PRINT)

يستخدم هذا الأمر بصفة خاصة عندما تكون هناك أوامر (@..SAY) يتم من خلالها عرض البيانات . فعند استخدام الأمر (SET DEVICE TO PRINT) يتم توجيه هذه البيانات الى الطابعة بدلا من الشاشة . وهذا يساعد على التحكم فى شكل ومكان البيانات على الورقة المطبوعة . حيث يمكن عن طريق كتابة الاحداثيات بعد (@) تحديد مكان الطباعة على الورقة كما يتم بالنسبة للشاشة تماما . ويسمى ذلك الطباعة المشكلة (Formatted) . ويجب ملاحظة أن هذا الأمر يؤدى الى توجيه البيانات الى الطابعة مع عدم ظهورها على الشاشة . كما يجب التنبيه الى استخدام الأمر (SET DEVICE TO SCREEN) بعد انتهاء الطباعة .

٢٣ - ٣ استخدام الأمر (SET PRINT ON)

يستخدم هذا الأمر مع الطباعة غير المشكلة (Unformatted) أى البيانات التى يتم عرضها من خلال الأوامر (DISPLAY, LIST, ?, ??) . ولكن لايستخدم مع الأمر (@...SAY) .

وهو يؤدى الى طباعة البيانات على الطابعة مع ظهورها على الشاشة فى نفس الوقت ، الا فى حالة استخدام الأمر (SET CONSOLE OFF) الذى يؤدى الى عدم ظهورها على الشاشة .

والأمر (SET PRINT ON) يسمح باضافة بعض الامكانيات الخاصة بالطباعة مثل طباعة الحروف المائلة (Italic) أو المضغوطة (Condensed) .

٢٣ - ٤ التحويل بين الشاشة والطابعة

كما سبق الايضاح ، فان استخدام الأمر (SET DEVICE TO PRINT) يؤدى الى توجيه السطور (SAY..@) الى الطابعة بدلا من الشاشة . وبالتالى لاتظهر هذه السطور على الشاشة أثناء الطباعة . ولذلك يمكن لمخطط البرامج استغلال الشاشة أثناء الطباعة فى عرض بعض الرسائل للمستخدم ، مثل اضافة ورقة جديدة للطابعة مثلا . وذلك عن طريق التحويل بين الشاشة والطابعة حسب الحاجة . ولتوضيح ذلك يمكن ملاحظة الأوامر التالية :

```
CLEAR
@ 10,10 SAY "Insert paper correctly"
?
WAIT SPACE(10) + "Press any key to begin printing"
SET DEVICE TO PRINT
        ---------
        ---------
        ---------      printing
        ---------
        ---------
SET DEVICE TO SCREEN
CLEAR
? CHR(7)
@ 10,10 SAY "Insert another piece of paper"
?
WAIT SPACE(10) + "and press any key to begin again"
SET DEVICE TO PRINT
```

ويلاحظ هنا أن البرنامج ينبه المستخدم الى وضع ورقة جديدة فى الطابعة بعد انتهاء صفحة الطباعة . وتم استخدام الجرس لتنبيه المستخدم عند انتهاء الصفحة عن طريق الدالة CHR(7) .

ولكن كيف يستطيع البرنامج تحديد نهاية صفحة الطباعة ؟ .

يستطيع البرنامج تنفيذ ذلك عن طريق الاحداثيات التى يتم كتابتها فى الأمر (SAY...@) . فعند كتابة رقم سطر أصغر من رقم سطر سابق تقوم الطابعة بالبداية من صفحة جديدة عند هذا السطر . فمثلا عند كتابة السطرين التاليين :

```
@ 10,10 SAY "-------"
@ 9,10  SAY "-------"
```

فان ذلك يؤدى الى القفز (Eject) الى ورقة جديدة ثم التحرك الى السطر رقم (٩) لكتابة السطر التالى . لذلك يمكن عن طريق البرنامج التحكم فى عدد السطور التى تظهر فى الورقة ، وتوقيت الانتقال الى الصفحة التالية .

ويمكن أيضا القفز الى الصفحة التالية باستخدام الأمر (EJECT) .

تحـــذيـر

الأمر (EJECT) يعمل بصورة جيدة عندما يكون قد سبق الطباعة من أول الورقة تماما .

٢٣ - ٥ تحديد الهامش الأيسر

يتم تحديد الهامش الأيسر أولا بتحديد مكان العمود رقم صفر كما يظهر على الورقة . ويمكن تنفيذ ذلك عن طريق كتابة البرنامج الصغير التالى :

```
SET DEVICE TO PRINT
@ 0,0 SAY 'Testing...'
EJECT
SET DEVICE TO SCREEN
```

ثم وضع الورقة على الطابعة وتنفيذ هذا البرنامج . ومن خلال ملاحظة بدء كتابة كلمة (Testing) على الطابعة يمكن تحديد مـكان الكتابة المقابل للعمود صفر على شاشة الحاسب .

كما يمكن استخدام الأمر (SET MARGIN TO) فى تغيير مكان الكتابة بالنسبة للمكان السابق تحديده . أى يمكن بواسطة هذا الأمر التحكم فى المسافة الكلية بين بداية الكتابة وبداية الورقة من اليسار .

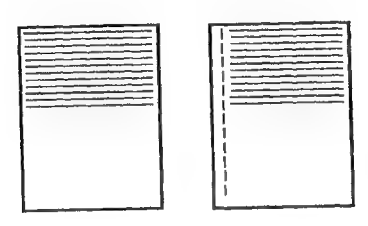
فمثلا عند كتابة الأمر

```
SET MARGIN TO 20
```

فى هذه الحالة يلاحظ التغيير فى شكل ورقة الطباعة كما هو موضح بالشكل .

ولتحديد الهامش العلوى (Top Margin) يمكن استخدام نفس البرنامـج السابق . حيث يمكن ملاحظة أول سطر فى الطباعة والذى يقابل السطر صفر على الشاشة . ولتحديد هذا الهامش يمكن اضافة عدة سطور خالية قبل بداية التقرير المكتوب على الشاشة وكذلك يمكن ترك عدة سطور خاليـــة أســـفل التقـــرير

حـــتى يتـــم التحكم في الهامش السفلي (Bottom Margin) .



شكل (٢٣ - ١)

٢٣ - ٦ طباعة السطر الأخير من التقرير

تحتوى الطابعة علي مخزن ذاكرة مؤقت (Buffer) يقوم بتخزين سطر الطباعة تخزينا مؤقتا قبل انتقاله الى ورقة الطباعة . ولايتم انتقال هذا السطر من المخزن المؤقت (Buffer)الا عند ادخال شفرة الادخال(Carriage Return Code). وهذه الشفرة هى كود الآسكي (13) . وفي الوضع العادي يتم ادخال هذه الشفرة بعد نهـــاية السطر على الشاشة عند الضغط على مفتاح الادخال . ولكن برنامج (DBase III+) يرسل هذه الشفرة في بداية السطر ، لذلك فان طباعة جميع سطور التقرير لايحدث فيها أى مشكلة ، أما السطر الأخير فهناك احتمال عدم طباعته على ورقة الطباعة . ويمكن التغلب على هذه المشكلة عن طريق استخدام الأمر (EJECT)، حـــيث أن هـــذا الأمـــر يـرســـل شفـــرة الادخـــال (Carriage Return Code) بالاضافة الى نقل الصفحة .

كما يمكن التغلب على هذه المشكلة أيضا من طريق كتابة السطور التالية فى نهاية التقرير :

```
SET PRINT ON
?? CHR(13)
SET PRINT OFF
```

أو كتابة السطور التالية :

```
SET DEVICE TO PRINT
@ 23,0 SAY CHR(13) && Sends carriage return code
SET DEVICE TO SCREEN
```

ملاحـظة

هذه المشكلة تظهر فقط عند طباعة التقرير من خلال البرنامج ولكنها لاتظهر عند طباعـة التقرير من خلال برنامج المساعد (Assistant) أو من خلال الأمر (CREATE/MODIFY REPORT) .

٢٣-٧ ادخال بعض المؤثرات الخاصة (Special Effects)

يمكن عن طريق ادخال شفرة تحكم (Control Code) الى الطابعة ، الحصول على مؤثرات خاصة تزيد من كفاءة الطباعة.وتبدأ شفرة التحكم (Control Code) في معظم الطابعات بشفرة مفتاح الهروب (Escape Key) وهي كود الآسكي (ASCII 27) ، ويليها أي شفرة أخرى . ولذلك تسمى شفرة التحكم عادة شفرة الهروب (Escape Code) . فمثلا لكي تستخدم الطابعة في طباعة حروف مائلة (Italic) يجب ادخال شفرة التحكم (ESC F) . وتتم كتابة ذلك في البرنامج كالآتي :

```
SET PRINT ON
?? CHR(27) + "F"
```

كما يمكن انهاء كتابة الحروف المائلة عن طريق كتابة الأوامر التالية في البرنامج :

```
SET PRINT ON
?? CHR(27) + "G"
```

ملاحـظة

يجب التأكد من اعادة الطابعة الى الوضع المبدئي (Default) بعد انتهاء استخدام المؤثر الخاص الذى تم ادخاله . واذا لم يحدث ذلك فان الطابعة ستظل تعمل بهذا التأثير الخاص .

٢٣ - ٨ تحديد مكان انتقال الصفحة (Page Break)

عندما تكون التقارير المطلوب طباعتها طويلة ، فان من المهم التحكم في طول صفحة الطباعة بحيث لاتزيد مثلا عن ٦٠ سطرا . كما يتم الانتقال الى الصفحة التالية آليا . ويمكن تنفيذ ذلك من خلال البرنامج التالى :

```
SET TALK OFF
STORE 61 TO tline
STORE 5 TO tcolumn
STORE 0 TO pagenum
USE Cadets INDEX name
GO TOP
SET DEVICE TO PRINT
DO WHILE .NOT. EOF()
   IF tline > 60
      STORE 1 TO tline
      STORE pagenum + 1 TO pagenum
      @ tline,tcolumn + 66 SAY "page" + STR(pagenum,3)
      @ tline + 1 , tcolumn + 66 SAY DATE()
      @tline + 4 ,tcolumn + 30 SAY 'Names and phone;
               numbers'
   ENDIF
   @ tline, tcolumn SAY TRIM(name) + '' + Phone
   SKIP
   STORE tline + 1 to tline
ENDDO
EJECT
SET DEVICE TO SCREEN
RETURN
```

ويلاحظ من هذا البرنامج أنه تم انشاء متغير الذاكرة (tline) ليمثل رقم السطر في الصفحة وتم ادخال القيمة (61) فيه . كما تم انشاء متغير الذاكرة (tcolumn) ليمثل رقم العمود الذي يمثل وضع الحرف على السطر عند كتابة الأمر (@...SAY) ، وتم ادخال القيمة (5) في هذا المتغير . كما تم انشاء متغير الذاكرة (Pagenum) ليمثل رقم الصفحة وتم ادخال القيمة صفر في هذا المتغير . ثم تم استخدام الأمر (IF) مع وضع الشرط (Tline>60) بعده .

فعندما يزيد عدد السطور عن ٦٠ سطرا يتم تنفيذ الأوامر التالية . وهذا يؤدي الى البدء من صفحة جديدة . كما يؤدي الى زيادة رقم الصفحة واحدا . واذا لم يصل عدد السطور الى ٦٠ سطرا ، يتم تنفيذ الأوامر بعد (ENDIF) ، والتي تؤدي الى كتابة البيانات المطلوبة ثم الانتقال الى سجل جديد باستخدام الأمر (SKIP) مع زيادة رقم السطر واحدا .

ويلاحظ في نهاية البرنامج استخدام الأمر (EJECT) للتاكد من طباعة السطر الأخير ، ثم اعادة الطابعة الى الوضع المبدئي (Default) .

ملاحظة

ما سبق ذكره في هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل DBase IV ، FoxBase ، FoxBase + ، FoxPro .

الباب الرابع والعشرون

التعامل مع بيئة الحاسب

تطلق كلمة بيئة الحاسب (Environment) للدلالة على المكونات الخارجية التى تتعامل معها وحدة التشغيل المركزية (Central Processing Unit) . وتقاس كفاءة البرنامج المكتوب للحاسب بقدرته على التعامل مع هذه المكونات والتحكم فيها بما يحقق أحسن استغلال لخصائصها الفنية . ويتضمن هذا الباب شرح الوسائل المختلفة للتحكم فى بيئة الحاسب (Environment) من خلال البرنامج . كما يتضمن استخدام بعض الدوال المتقدمة (Advanced Functions) لتحقيق أحسن استغلال لهذه المكونات .

٢٤ - ١ التعامل مع القرص

فى العادة يتم فتح ملف قاعدة البيانات (DBase File) من خلال البرنامج . ولكن فى بعض الأحيان قد يتطلب الأمر أن يدخل المستخدم اسم الملف المطلوب استخدامه خصوصا فى البرامج التى تتصف بالكفاءة والقدرة على التعامل مع ملفات مختلفة وحقول بيانات مختلفة . والبرنامج يتيح للمستخدم استخدام الداله FILE() التى يستطيع المستخدم من خلالها ادخال اسم الملف المطلوب .

وهذه الدالة تعطى القيمة صحيح (True) أو غير صحيح (False) حسب وجود الملف المكتوب بين القوسين فى القرص أو عدم وجوده . مع ملاحظة أن اسم الملف يكون متضمنا الامتداد (Extension) . ويمكن من خلال البرنامج اضافة هذا الامتداد (DBF.) حتى يستطيع المستخدم ادخال اسم الملف دون أى ارتباك نتيجة عدم معرفة امتداد الملف .

والبرنامج التالى يوضح استخدام هذه الدالة :

```
mfile = SPACE(8)
CLEAR
@ SAY 'Enter the file name:' GET mfile PICTURE '@!'
READ
mfile = LTRIM(TRIM(mfile)) + '.DBF'
IF FILE(mfile)
      -----
      -----   comands
      -----
ENDIF
```

وهذا البرنامج يقوم بانشاء متغير ذاكرة (mfile) طوله ٨ حروف لادخال اسم الملف فيه . ثم يقوم البرنامج بعرض رسالة للمستخدم ليدخل اسم الملف المطلوب ثم يقوم بالتخلص من المسافات الخالية (Spaces) ويضيف الامتداد (DBF) الى الاسم .

ويستخدم الأمر (IF) لاختبار وجود هذا الملف على القرص فاذا وجده ينفذ الأوامر التالية .

ويجب ملاحظة أن هذا البرنامج يقوم بالبحث خلال وحدة الأقراص الحالية (Current Drive). فاذا أريد ادخال مسار معين (Path) يتم ادخال هذا المسار أولا فى متغير ذاكرة ، وذلك كالآتى مثلا :

```
STORE 'C:\ DBase\ Cadets\' TO Path
IF FILE(path + mfile)
      ------
      ------
      ------  commands
      ------
      ------
 INDIF
```

كما يمكن استخدام الداله DBF() فى معرفة اسم ملف قاعدة البيانات المفتوح فى منطقة العمل الحالية (Current Work Area) . وكذلك تستخدم الدالة NDX() فى معرفة أسماء ملفات الفهرس المستخدمة فى منطقة العمل الحالية .

وحيث أن برنامج (DBase III+) يسمح بفتح حتى ٧ ملفات فهرس مع كل ملف قاعدة بيانات . لذلك فان الداله NDX() تعطى اسم الملف حسب الرقم الذى يتم ادخاله بين القوسين . فمثلا الأمر التالى يعطى اسم ملف الفهرس الرئيسى :

```
? NDX(1)
```

وتستخدم الداله FIELD() في تحديد اسم أى حقل داخل ملف قاعدة البيانات من خلال البرنامج . وهذه الدالة تقبل أى رقم من ١ الى ١٢٨ بين القوسين . وهذا الرقم يمثل ترتيب الحقل المطلوب داخل الملف . فمثلا الأمر التالى يوضح اسم أول حقل فى الملف :

```
? FIELD(1)
```

وهذه الدوال تساعد مخطط البرامج على كتابة برنامج يستطيع التعامل مع ملفات مختلفة بتركيب (Structure) مختلف لكل ملف .

وهناك دالة أخرى تستخدم فى تحديد آخر تاريخ تم تعديل الملف فيه وهى الدالة LUPDATE() . وهذه الدالة تكون مفيدة للتأكد من عدم التكرار (Duplication) لبعض العمليات مثل التجميع اليومى للكميات (Daily Totaling).

٢٤ - ٢ تحديد حجم الملف وحجم القرص المستخدم

يمكن من خلال البرنامج التحكم فى حجم الملف المستخدم وتنبيه المستخدم عندما يزيد هذا الحجم بدرجة كبيرة حتى يقوم بالتخلص من بعض السجلات التى قد لاتكون هناك حاجة اليها . وتستخدم لذلك الدالة RECCOUNT() . هذه الدالة تعطى العدد الكلى للسجلات داخل الملف . والبرنامج التالى يوضح استخدامها :

```
USE Cadets
IF RECCOUNT() > 2000
     ? CHR(7)
     @ 10,10 SAY 'Please delete some records'
ENDIF
```

كما يمكن أيضا تحديد المساحة التخزينية التي يحتاجها الملف ، وذلك باستخدام الدالة RECSIZE() . هذه الدالة تعطى المساحة التخزينية التى يحتاجها السجل الواحد من الملف . أى أنها تعطى مجموع المساحات التخزينية للحقول بالحروف (Bytes). وحتى يمكن تحديد المساحة التخزينية للملف يتم ضرب عدد السجلات فى المساحة التخزينية للسجل الواحد كالآتى مثلا :

```
STORE RECCOUNT * RECSIZE TO size
```

وهذا فى الواقع لايعطي المساحة التخزينية الكلية للملف حيث أن هناك مساحة أخرى تستخدم كمعلومات تقديمية (Header Information). هذه المعلومات التقديمية تساعد على تتبع أسماء الحقول وأطوالها وأنواعها . ولذلك يجب اضافة المساحة التخزينية لهذه المعلومات حتى يتم حساب المساحة التخزينية الكلية بدقة .

وحتى يتم حساب المساحة التخزينية للمعلومات التقديمية (Header Information). يجب أولا معرفة عدد الحقول داخل السجل وذلك عن طريق الأوامر التالية :

```
USE Cadets
numfields = 0
null = ""
DO WHILE null < FIELD(numfields + 1)
     numfields = numfields + 1
ENDDO
USE
```

والحلقة التكرارية فى هذا البرنامج تؤدى الى زيادة عدد الحقول بواحد طالما كان أسم الحقل أكبر من السلسلة الحرفية الخالية (null) . وحيث أن السلسلة الحرفية

الخالية قيمتها صفر ، فان أي حروف موجودة في اسم الحقل تؤدى الى زيادة عدد الحقول بواحد . وهكذا يعطى هذا البرنامج العدد الكلى للحقول في الملف .

ولحساب المساحة التخزينية للمعلومات التقديمية تستخدم العلاقة التالية :

```
header = (32 * numfields) + 34
```

ومن هذا يمكن حساب المساحة التخزينية الكلية للملف عن طريق العلاقة التالية :

```
totalsize = size + header + 1
```

حيث (Size) هو الحجم الذى سبق تحديده عن طريق ضرب عدد السجلات RECCOUNT() في المساحة التخزينية للسجل RECSIZE(). كما أن الرقم (١) تم اضافته في المعادلة ليمثل علامة نهاية الملف (End of file marker) والتى تحتل حرفا واحدا (One byte) .

ومعرفة المساحة التخزينية الكلية للملف تكون في بعض الأحيان عملية مهمة جدا . فمثلا عندما يراد عمل فرز للملف (Sorting) ، فالمعروف أن عملية الفرز تؤدى الـــى انشاء ملف آخر غير الملف الأصلي . أى أنه يلزم أولا التأكد أن المساحة التخزينية للقرص تزيد عن ضعف المساحة التخزينية لملف قاعدة البيانات . لذلك يمكن استخدام البرنامج التالى لتنفيذ هذه العملية .

```
USE Cadets
IF DISKSPACE < totalsize * 2
  ? CHR(7)
  @ 10,10 SAY "Not enough room in disk to sort this"+;
"file"
ELSE
  SORT ON grade DESCENDING TO Temp
ENDIF
```

ويؤدى هذا البرنامج الى اختبار المساحة التخزينية للقرص عن طريق الدالة DISKSPACE() . فاذا كانت أقل من ضعف المساحة التخزينية لملف قاعدة البيانات يتم تنبيه المستخدم حتى لايقوم باجراء عملية الفرز . ويلاحظ هنا استخدام الدالة CHR(7) لتنبيه المستخدم عن طريق تشغيل الجرس (Bell) .

ويمكن استخدام الداله DISKSPACE() في عمل نسخة احتياطية من ملف قاعدة البيانات الكبير والمخزن على القرص الصلب وذلك بنسخ الملف في مجموعة من الأقراص المرنة . ويمكن تنفيذ ذلك عن طريق نسخ مجموعة من السجلات الى كل قرص مرن حسب السعة التخزينية لهذا القرص . ويتم ذلك عن طريق البرنامج التالى :

```
USE Cadets
SET DEFAULT TO B
DO WHILE .NOT. EOF()
    WAIT"insert new disk in drive B,and press a key"
    COPY NEXT(DISKSPACE()-(headersize))/RECSIZE();
          TO Backup
      SKIP
ENDDO
USE
```

ويلاحظ من خلال الحلقة التكرارية أن البرنامج يستمر في نسخ السجلات واحدا تلو الآخر الى القرص المرن الموجود في وحدة الأقراص (B) طالما كانت النسبة بين المساحة التخزينية للقرص ناقصا المساحة التخزينية للمعلومات التقديمية (Header Information) وبين المساحة التخزينية للسجل الواحد أكبر من (١) . وعندما تصبح هذه النسبة أقل من واحد (وهذا يعنى أن المساحة التخزينية المتبقية على القرص لا تكفي لتخزين سجل) ، فان السجل التالى لايتم نسخه ويتوقف تنفيذ الحلقة التكرارية حتى يقوم المستخدم بوضع قرص جديد لتخزين مجموعة أخرى من السجلات . وهكذا .

٢٤ - ٣ مسح وتغيير اسم الملف

يمكن من خلال البرنامج اعطاء المستخدم الفرصة لمسح الملف ، أو تغيير اسمه حسب الحاجة . ويجب في هذه الحالة اضافة الامتداد الى اسم الملف . ويمكن استخدام الطريقة السابق شرحها باستخدام الدالة FILE(). كما يمكن استخدام البرنامج التالى لمساعدة المستخدم على كتابة الاسم متضمنا الامتداد .

```
File = SPACE(12)
@ 10,10 SAY 'Enter name of file to delete'
@ 12,10 SAY 'including file extension'
@ 14,5 GET file PICTURE '!!!!!!!! . !!!'
READ
IF FILE(file)
     ERASE & file
ELSE
     ? CHR(7)
@ 18,10 SAY 'There is no file by that name'
```

وهذا البرنامج يجبر المستخدم على اضافة الامتداد (Extension) فى نهاية اسم

الملف . ويلاحظ هنا استخدام الحرف (&) الذي يسمى التعويض بالماكرو (Macro Substitution) ، والذى يتم من خلاله وضع اسم الملف الذى أدخله المستخدم بعد الأمر (ERASE) .

ويمكن استخدام الأمر (RENAME) أيضا لتغيير اسم الملف . ويمكن فى هذه الحالة استخدام نفس البرنامج وذلك عن طريق كتابة الأمر التالى بدلا من الأمر (ERASE).

```
RENAME & file TO Cadets
```

٢٤ - ٤ تعديل تركيب الملف (Modifying Structure)

عندما يراد تعديل تركيب ملف قاعدة البيانات فان من المهم السيطرة على هذه العملية من خلال البرنامج . وذلك لأن تعديل هذا التركيب قد يؤدى الى توقف البرنامج عندما يتم بواسطة المستخدم دون توجيه أو اشراف من البرنامج .

ولتنفيذ ذلك يتم أولا نسخ تركيب الملف (Structure of dbase file) فى ملف مؤقت (Temporary) مع استخدام الاختيار (Extended) . هذا الاختيار يؤدى الى تحويل تركيب الملف الى مجموعة من السجلات . أي أن كل حقل يصبح سجلا فى الملف الجديد . وهذا يساعد على مسح أو تعديل أى حقل بنفس الطريقة التي يتم بها تعديل أو مسح أى سجل في ملف قاعدة البيانات . وعند الانتهاء من ادخال التعديلات المطلوبة يتم انشاء ملف جديد بالتركيب المطلوب . وذلك عن طريق الأمر (CREATE FROM) .

فمثلا لتنفيذ هذه العملية على الملف (Cadets) الذى يحتوى على الحقول التالية :

Field	Field Name	Type	Width	Dec
1	name	Character	30	
2	address	Character	30	
3	age	Numeric	4	1
4	grade	Numeric	5	2

يتم أولا نسخ هذا التركيب (Structure) في ملف مؤقت (Temporary) يتم تسميته (Temp) مثلا مع استخدام الاختيار (EXTENDED) .

فى هذه الحالة يتم تحويل الحقول الى سجلات ويكون كل سجل منها مكونا من أربعة حقول هى اسم الحقل (Field_name) ، ونوع الحقل (Field_Type) ، وطول الحقل (Field_Len) ، وعدد الأرقام العشرية (Field_Dec).

ويصبح حقل الاسم هو السجل الأول فى قاعدة البيانات وحقل العنوان هو السجل الثانى .. وهكذا .

فمثلا لكى يتم مسح حقل الدرجة (Grade) يتم كتابة الأمر التالى فى البرنامج :

```
DELETE RECORD 4
```

كما يمكن تعديل طول أى حقل مثلا باستخدام الأمر التالى :

```
REPLACE Field_Len WITH 35 FOR Field_name = "address"
```

وذلك يؤدى الى تعديل طول حقل العنوان من ٢٠ الى ٣٥ حرفا .

وبعد الانتهاء من ادخال جميع التعديلات المطلوبة على الحقول من خلال البرنامج يتم استخدام الأمر (PACK) لتخزين هذه التعديلات . ثم يتم استخدام الأمر (CREATE FROM) لاعادة الملف المؤقت (Temp) من الصورة الممتدة (EXTENDED) الى الصورة الطبيعية . أى تحويل السجلات الى حقول مرة ثانية . وذلك بكتابة الأمر التالى :

```
CREATE New FROM Temp
```

ويؤدى هذا الى انشاء ملف جديد اسمه (New) بالتركيب المطلوب . ويجب ملاحظة أن التعديلات التى تم ادخالها تتعلق بتركيب قاعدة البيانات فقط ولكن السجلات مازالت كما هى . ولكى يتم تعديل السجلات على التركيب الجديد ، يتم استخدام الأمر التالى :

```
APPEND FROM Cadets
```

ويجب التأكد قبل ذلك من فتح الملف (New) باستخدام الأمر (USE) . ثم يتم اغلاق الملف (New) الذى يتم تغيير اسمه بعد ذلك بواسطة الأمر (RENAME) الى (Cadets) مرة ثانية .

٢٤ - ٥ خطوات انهاء البرنامج

قبل انتهاء البرنامج ، هناك مجموعة من الأوامر التى يتم كتابتها لاغلاق الملفات المفتوحة ، ومسح متغيرات الذاكرة ثم العودة الى الوضع المبدئى للبرنامج . وهى تختلف حسب نوع البرنامج وهل هو فرعى أو رئيسى .

ففى حالة البرنامج الفرعى يتم الرجوع الى البرنامج الذى قام باستدعاء هذا

البرنامج الفرعي بواسطة الأمر (RETURN) . وهذا الأمر يقوم بمسح جميع متغيرات الذاكرة الخاصة (Private) ، ولكنه لايمسح المتغيرات العامة (Public) . ولكي يتم مسح المتغيرات العامة يستخدم الأمر (CLEAR MEMORY) . ولكن يجب الحذر عند استخدام هذا الأمر لأنه يمسح كل الذاكرة المؤقتة . وقد تكون بعض البرامج الأخرى في حاجة الى بعض المتغيرات المخزنة بها .

٢٤ - ٥ - ١ اغلاق الملفات

من المهم جدا قبل انهاء البرنامج التأكد من اغلاق ملفات قواعد البيانات بصفة خاصة . لأن عدم اغلاقها قد يؤثر في تكامل قاعدة البيانات (Integrity) . ويستخدم الأمر التالى لاغلاق ملفات قواعد البيانات :

```
CLOSE DATABASES
```

وهذا الأمر لا يغلق ملفات قواعد البيانات فقط ولكنه يغلق أيضا ملفات الفهرس (Index) والتشكيل (Format) وباقى الملفات المرتبطة بها .

٢٤-٥-٢ العودة الى البيئة المبدئية (Default Environment)

من المهم أيضا قبل انهاء البرنامج التأكد من العودة الى الأوضاع المبدئية (Default) لبرنامج (DBase III+) حتى لاتؤثر الأوضاع التى تم ادخالها على أي برامج أخرى يتم كتابتها . ويتم ذلك عن طريق تعديل كل وضع للأمر (SET) الى عكس الحالة التى تم ادخالها . فمثلا عند كتابة الأمر (SET TALK OFF) فى بداية البرنامج ، يتم كتابة الأمر (SET TALK ON) فى نهاية البرنامج .. وهكذا .

والسطور التالية توضح الأوامر التى يتم استخدامها عادة فى نهاية معظم البرامج :

```
SET TALK ON
SET ESCAPE ON
SET BELL ON
SET HEADING ON
SET HELP ON
SET SEFETY ON
SET STATUS ON
CLEAR ALL
CLEAR
RETURN
```

ملاحظة

ما سبق ذكره في هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل (DBase IV) ، (FoxBase +) ، (FoxPro) .

الباب الخامس والعشرون

استخدام وسائل أكثر تقدما

More Advanced Techniques

٢٥ - ١ استخدام الدالة (IIF)

الدالة (IIF) هي صورة أخرى للأمر (IF) . وهذه الدالة يتم عن طريقها كتابة الشـــرط وجواب الشرط على نفس السطر ، حيث لاتكون هناك حاجة لكتابة الأمـر (ELSE) أو الأمر (ENDIF) . ويتم ذلك عن طريق كتابة الشرط وجوابه بين قوسين بعد الدالة (IIF) مع فصلهما بفاصلة (,) . فمثلا الأوامر التالية تستخدم الأمر (IF).

```
IF sex = 'F'
mname = 'Ms.' + name
ESLE
mname = 'Mr.' + name
ENDIF
```

فعند استخدام الدالة (IIF) يتم اختصار هذه السطور الى السطر التالى :

```
mname = IIF(sex = 'F', 'Ms.' , 'Mr.') + name
```

ويعني هذا اختبار قيمة الحقـل (sex) أى الجنس فاذا كان (F) أى (Female) أو أنثى ، يتم كتابة الحروف (.Ms) قبل الاسم (name) واذا كان غير ذلك يتم كتابة الحروف (.Mr) قبل الاسم . ثم يتم تخزين النتيجة فى التغير (mname) .

واختصار الســـطور بهذه الطريقة يؤدى الى سهولة تصحيح البرنامج بالاضافة الى سرعة تنفيذه .

٢٥ - ٢ استخدام ملف الخطوات (Procedure File)

عند تصميم البرنامج ، فان مخطط البرامج يكتشف أن هناك بعض البرامج الفرعية (Modules) التي تستخدم بكثرة فى البرنامج . وفى كل مرة يتم فيها تنفيذ أى برنامج من هذه البرامج الفرعية ، يستخدم الأمر (DO) فى الانتقال من البرنامج الرئيسى الى البرنامج الفرعى كما سبق الايضاح .

وحيث أن البرنامج الفرعي يكون دائما مخزنا فى ملف منفصل ، فان هذا الملف المنفصل يكون موجودا على القرص حتى يتم استدعاؤه . لذلك فان البرنامج يذهب الى القرص فى كل مرة يراد فيها فتح ملف برنامج فرعى . وهذا قد يستغرق وقتا طويلا خصوصا عند تعدد هذه البرامج الفرعية .

ولذلك يتيح برنامج (+DBase III) لمخطط البرامج استخدام ما يسمى

بملف الخطوات (Procedure File) . هذا الملف يحتوى على برامج صغيرة (Modules) تبقى في الذاكرة المؤقتة طوال فترة تشغيل البرنامج . وبالتالى فان البرنامج لا يحتاج الى الذهاب الى القرص عدة مرات لتحميل هذه البرامج .

وملف الخطوات يتم انشاؤه وتعديله باستخدام الأمر (MODIFY COMMAND) . لذلك يتم اضافة الامتداد (Prg.) الى اسمه آليا . ويتم كتابة كل برامج الخطوات (Procedures) داخل هذا الملف ، مع ملاحظة أن كل برنامج من هذه البرامج يبدأ بالأمر (PROCEDURE) يليه اسم هذا البرنامج ، ثم تأتي باقى أوامر البرنامج . والسطر الأخير من كل برنامج يحتوى على الأمر (RETURN) .

ملاحظة

بدلا من كتابة ملف الخطوات (Procedure File) مباشرة يمكن كتابة كل برنامج فرعى منفصلا ، وذلك باستخدام الأمر (MODIFY COMMAND) يليه اسم البرنامج الفرعى المطلوب كتابته . ثم يتم اختبار كل برنامج فرعي والتأكد أنه يحقق الهدف منه . وعند الانتهاء من جميع البرامج الفرعية ، يتم انشاء ملف الخطوات (Procedure File) باستخدام الأمر (MODIFY COMMAND) أيضا . كما يتم نقل كل برنامج فرعى تم انشاؤه الى ملف الخطوات عن طريق الضغط على مفتاحى (Ctrl-KR) . حيث يسأل البرنامج عن اسم الملف المراد نقله الى ملف الخطوات . ويمكن اتباع نفس الطريقة فى اضافة أى برنامج فرعى جديد الى ملف الخطوات (Procedure File) .

وعند الحاجة الى تشغيل ملف الخطوات مع البرنامج الرئيسى يستخدم الأمر (SET PROCEDURE TO) مع كتابة اسم ملف الخطوات بعده وذلك فى بداية البرنامج ثم يتم تنفيذ أى برنامج فرعى داخل ملف الخطوات باستخدام الأمر التالى مثلا :

```
DO Report
```

مع ملاحظة أنه لايتم فتح أكثر من ملف خطوات واحد فى كل مرة . ولكن يمكن اغلاق الملف المستخدم وفتح ملف خطوات آخر فى أى وقت .ولكى يتم فتح ملف خطوات آخر يستخدم الأمر (SET PROCEDURE TO) ثم كتابة اسم الملف المراد فتحه . وفى هذه الحالة يتم اغلاق الملف السابق وفتح الملف الجديد .

كما يمكن اغلاق ملف الخطوات دون فتح أى ملف آخر ، وذلك بكتابة الأمر (SET PROCEDURE TO) أو الأمر (CLOSE PROCEDURE) دون كتابة أى اسم بعده . ويمكن أن يحتوى ملف الخطوات الواحد على مالايزيد عن ٣٢ برنامجا فرعيا (Procedure) .

ملاحـــظة

يمكن لبرنامج خطوات أن يستدعى برنامج خطوات آخر فى نفس ملف الخطوات المفتوح . وذلك بكتابة الأمر (DO) وبعده اسم برنامج الخطوات المطلوب .

والسطور التالية توضح انشاء ملف الخطوات الذى يتم تسميته (Test) مثلا :

```
PROCEDURE Proc1
     ? "This is a message from Proc1"
RETURN
PROCEDURE Proc2
     ? "Greetings From Proc2"
     DO Proc3
RETURN
PROCEDURE Proc3
     ? "greetings From Proc3."
RETURN
```

ولتحميل هذا الملف ، يتم كتابة الأمر التالى :

```
SET PROCEDURE TO Test
```

كما يمكن تشغيل البرنامج الأول والثانى كالآتى :

```
DO Proc1
DO Proc2
```

ويلاحظ فى هذه الحالة ظهور الآتى على الشاشة :

```
This is a message from Proc1.
Greetings From Proc2.
Greetings From Proc3.
```

ويلاحـــظ هنا ظهور الرسالة الخاصة ببرنامج الخطوات (Proc3) رغم عدم استدعائه . وذلك لأن البرنامج (Proc2) يستدعيه من داخله .

٢٥ - ٢ اخفاء المتغير العام (Public Variable)

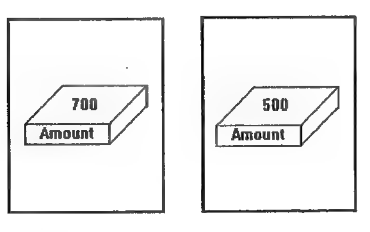
كما سبق الايضاح فان المتغير العام يؤثر فى جميع البرامج حتى اذا تم انشاؤه داخل برنامج فرعي . وذلك عكس المتغير الخاص (Private) الذى لايؤثر الا على البرنامج الفرعي الذى تم انشاؤه خلاله أو البرامج الفرعية المتفرعة منه . ولتوضيح ذلك نفرض أن المتغير (amount) تم انشاؤه داخل أحد البرامج الفرعية . وكانت قيمته داخل هذا البرنامج الفرعي (500) مثلا . فاذا تم استخدام المتغير (amount) فى أى برنامج آخر ، فان هذا المتغير يأخذ نفس القيمة (500) .

وفى بعض الأحيان يراد استخدام المتغير العام فى أحد البرامج الفرعية دون أن يتأثر بالقيمة المخزنة فى هذا المتغير أى استخدامه كمتغير خاص لهذا البرنامج الفرعي فقط . فى هذه الحالة يتم اخفاء المتغير العام عن هذا البرنامج الفرعى . ويتم هذا الاخفاء باستخدام الأمر التالى :

```
PRIVATE amount
```

فعند تشغيل هذا البرنامج الفرعي يظل المتغير (amount) متغيرا خاصا بالنسبة لهذا البرنامج والبرامج الفرعية المتفرعة منه . أى أن قيمته لاتتأثر بالمتغير العام (amount) الموجود فى البرنامج الرئيسى .

وعند انتهاء البرنامج الفرعى ، يعود المتغير (amount) الى حالته الأولى ، أى يصبح متغيرا عاما .



شكل (٢٥ - ١)

ويلاحظ من الشكل (٢٥ - ١) أن المتغير (amount) يأخذ القيمة (500) في نهاية تنفيذ البرنامج ، دون أن يؤثر ذلك في قيمة نفس المتغير في البرنامج الرئيسي .

٢٥ - ٤ ادخال المعاملات (Parameter Passing)

كما سبق الايضاح فان استخدام البرامج الفرعية (Modules) أو برامج الخطوات (Procedures) يتيح لمخطط البرامج استخدام البرنامج الفرعي عدة مرات وفى أماكن مختلفة من البرنامج . ولكن في بعض الأحيان يراد استخدام نفس البرنامج الفرعي مع تعديل بعض القيم المستخدمة به .

فمثلا قد يراد أحيانا رسم مستطيل على الشاشة حول البيانات المختلفة . ولكن قد يراد رسم هذا المستطيل في أماكن مختلفة من الشاشة . وفي هذه الحالة تسبب كتابة برنامج فرعي منفصل لكل مستطيل استهلاكا للوقت والذاكرة المتاحة . لذلك يتيح البرنامج استخدام المعاملات (Parameters) التى يتم ادخالها الى البرنامج الفرعي من البرنامج الذى قام باستدعائه .

ويتم تنفيذ ذلك عن طريق كتابة البرنامج مع استخدام متغيرات تمثل القيم المطلوب استخدامها كمعاملات . فمثلا يمكن كتابة برنامج الخطوات (Procedure) الذى يتم تسميته (Box) كالآتى :

```
PARAMETERS beginrow, begincol, endrow, endcol
CLEAR
@ beginrow, begincol TO endrow, endcol DOUBLE
RETURN
```

ويلاحظ فى هذا البرنامج كتابة الأمر (PARAMETERS) يليه أسماء المعاملات المتغيرة (Beginrow) ، (Begincol) ، (Endrow) ، (Endcol) .

وعندما يراد رسم هذا المستطيل في مكان محدد على الشاشة مثلا يستخدم الأمر التالى :

```
DO Box WITH 10,9,23,75
```

ويؤدى هذا الى استدعاء البرنامج الفرعي (Box) مع ادخال المعاملات المطلوبة مكان المعاملات المتغيرة (beginrow) ، (begincol) ، (endrow) ، (endcol) بنفس ترتيبها .

ويمكن استخدام هذا البرنامج (Box) فى أى مكان آخر من البرنامج مع تغيير هذه المعاملات .

٢٥ - ٥ استخدام الأمر (RUN)

يمكن تشغيل برامج أخرى من خلال البرنامج مثل برامج نظام التشغيل الساكنة في الذاكرة (Memory Resident Programs)، وذلك باستخدام الأمر(RUN) . كمـا يمكــن استخدام الأمر (!) بدلا من الأمر (RUN) حيث أنه يؤدى نفس الغرض . وهذه البرامج يتم تشغيلها بكتابة اسم البرنامج مثل الأمر (FORMAT) الذي يستخدم في تجهيز قرص جديد ، والأمر (CHKDSK) الذى يستخدم في اختبار القرص واكتشاف أى قطاعات غير سليمة (Bad Sectors) . ويجب ملاحظة أن الأمر (RUN) يتطلب ذاكرة مؤقتة (RAM) تزيد عن ٢٥٦ كيلوبايت . كما يجب أن يكون الملف (Command.com) موجـــودا فـــى الفهرس الرئيسى (Root Directory) .

ويستخــدم الأمر (RUN) بصفة خاصة فى ادخال تاريخ اليوم الحالى الى البرنامج . ويتم ذلك عن طريق ادخال تاريخ اليوم فى متغير ذاكرة ثم استخدام الأمر DATE فى ادخال هذا التاريخ .

```
Today = '7/5/89'
RUN DATE & Today
```

ويؤدى هذا الى ادخال التاريخ الموجود فى متغير الذاكرة (Today) لكى يصبح هو تاريخ اليوم الحالى .

٢٥ - ٦ نظام التشغيل

عند كتابة برنامج ويراد زيادة كفاءة وانتقالية هذا البرنامج (Portability) ، أي قدرته على العمل على نظم تشغيل متعددة ، فان ذلك يتطلب أن يعرف البرنامج نظام التشغيل الذى يعمل عليه ، حتى يتم ادخال تجهيز معين للبرنامج حتي يعمل على هذا النظام . ويمكن لمخطط البرامج استخدام الداله ()OS في تنفيذ ذلك . وهذه الدالة عند كتابتها في البرنامج ، فانها تعطي سلسلة حرفية (String) تمثل اسم هذا النظام . ويمكن ملاحظة ذلك من خلال البرنامج التالى :

```
STORE OS() TO opsys
IF SUBSTR(opsys, 1, 4) = 'UNIX'
DO Setunix
ENDIF
```

وهذا البرنامج يقوم أولا باختبار نظام التشغيل المستخدم فاذا وجد أنه النظام

(UNIX) فانه ينفذ البرنامج (Setunix) الذى يؤدى الى تجهيز البرنامج للعمل على نظام التشغيل يونيكس .

٢٥ - ٧ التعويض بالماكرو (Macro Substitution)

تستخدم الدالة (&) للتعويض عن القيمة الحرفية لمتغير حرفي فى أى مكان داخل البرنامج . وهذه الدالة مهمة جدا لزيادة كفاءة البرنامج وسرعته . وهي تكون ضرورية مع بعض الأوامر مثل الأمر (FIND) مثلا . ولتوضيح وظيفة هذه الدالة يمكن دراسة الأوامر التالية :

```
STORE "Tarek" TO mname
USE Cadets INDEX Name
FIND & mname
```

فى هذه الحالة يقوم البرنامج بالبحث عن السلسلة الحرفية (Tarek) المخزنة فى المتغير (mname) كما يمكن ادخال الماكرو داخل سلسلة حرفية أخرى كما يتضح من الأوامر التالية :

```
STORE "Hasan" TO mname
STORE "Hello & mname" TO greeting
? greeting
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
Hello Hasan
```

كما يمكن اضافة أى حروف أخرى الى الماكرو وذلك بكتابة النقطة (.) وبعدها هذه الحروف كالآتى مثلا :

```
STORE "Hasan" TO mname
STORE "Hello & mname . ein" TO greeting
? greeting
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
Hello Hasanein
```

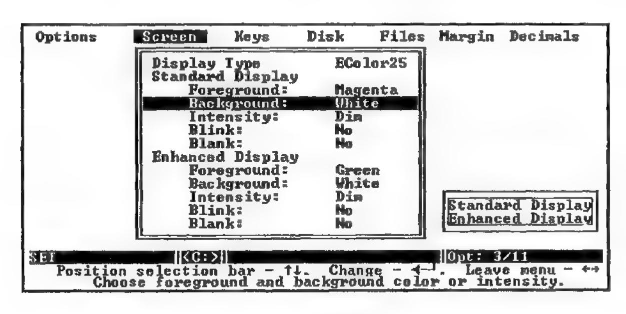
ويلاحظ هنا أنه تم اضافة الحروف ein الى الاسم Hasan .

٢٥ - ٨ التحكم فى الألوان (Colors)

يمكن الحصول على الألوان المطلوب ظهورها أثناء تنفيذ البرنامج . ويتم ذلك عن طريق كتابة الأمر (SET) عند مشيرة النقطة (Dot Prompt) . يلاحظ في هذه الحالة ظهور قوائم التجهيز .

وعن طريق تحريك المؤشر الضوئي (Highlight) أعلى الشاشة الى الاختيار (Screen) والضغط على مفتاح الادخال تظهر قائمة بالألوان المطلوب اختيارها وذلك بالنسبة للكتابة (Foreground) والخلفية (Background) .

ويلاحظ أن هناك قائمة خاصة بالشاشة (Standard Display) وقائمة أخرى خاصة بالأعمدة الضوئية التى تظهر خلال الشاشة (Enhanced Display) . كما أن هناك قائمة خاصة بالحدود (Borders) التى تظهر حول الشاشة . كما يلاحظ وجود اختيار لشدة اللون (Intensity) في جميع الحالات . وشدة اللون تعنى اذا كان مضيئا (Bright) أو معتما (Dim) . كما أن هناك اختيار اللون المتلألىء (Blinking) أو غير المتلألىء . انظر الشكل (٢٥ - ٢)



شكل (٢٥ - ٢)

وهذه الألوان تعطى شاشات الادخال وشاشات عرض البيانات شكلا جذابا ومثيرا .

وقد يريد مخطط البرامج في بعض الأحيان التحكم في هذه الألوان من خلال البرنامج . حيث أن ذلك يتيح له تنبيه المستخدم الى أهمية بعض المعلومات واعطاء التأثير المطلوب . ويتم ذلك من خلال الأمر (SET COLOR ON) للتحويل من

الشاشة- الأحـادية اللـون (Monochrome Monitor) الى الشاشة الملونة (Colored Monitor) ثم يتم اختيار الألوان بواسطة الأمر (SET COLOR TO) ، ثم كتابة سلسلة حرفية (String) تمثل الألوان المطلوبة . وهذه السلسله تنقسم الى أربعة أجزاء ، الجزء الأول يمثل لون الشاشة ، والجزء الثاني يمثل لون الأعمدة الضوئية (Enhanced) ، والجزء الثالث يمثل الحدود (Borders) والحزء الرابع يمثل الخلفية (Background). ويتم تمثيل كل لون بحرف أو حرفين حسب الجدول المبين .

اللـون	الحروف
احمر	R
بنفسجي	RB
بني	GR
أبيض	W
أسود	N
أزرق	B
أخضر	G
سماوي	BG
خالي	X

ولتوضيح ذلك يمكن كتابة الأمر التالي :

```
SET COLOR TO GR+/B , W/R, GR
```

وهذا يؤدي الى تحديد اللون الأصفر للكتابة في الشاشة على خلفية زرقاء . كما يؤدي الى تحديد لون الأعمدة الضوئية (Enhanced Video) ليكون لون الكتابة أبيض على خلفية حمراء . كما يحدد لون الحدود (Borders) ليكون بنيا .

ويلاحظ اضافة علامة + مع الحروف (GR) لتحويل اللون من بني الى أصفر ، حيث أن علامة (+) تؤدي الى زيادة شدة اللون (Intensity). كما يمكن اضافة العلامة (*) الى أي لون لتحويل اللون الى الوضع المتلالىء (Blinking). ويستخدم الحرف (X) في بعض الحالات ليجعل الكتابة بنفس لون الخلفية . ويفيد ذلك عند كتابة البرامج التي تستخدم في ادخال كلمة مرور للبرنامج (Password) . حيث يتطلب ذلك كتابة كلمة المرور بحيث لاتظهر أمام أي شخص موجود أمام الحاسب . لذلك يتم كتابتها بنفس لون الخلفية .

٢٥ - ٩ استخدام الاختصارات فى كتابة الأوامر

يتيح برنامج (+DBase III) لمخطط البرامج كتابة الأوامر بطريقة مختصرة تساعد على توفير الوقت والجهد المستهلك في كتابة البرنامج . حيث يمكن كتابة الأربعة حروف الأولى فقط من كل أمر ، وذلك كالآتى مثلا :

```
MODI COMM
```

وذلك بدلا من الأمر (MODIFY COMMAND) .

وينطبق هذا على أى أمر من أوامر برنامج (+DBase III) .

ملاحظة

ما سبق ذكره في هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل (DBase IV) ، (+ FoxBase) ، (FoxPro) .

الباب السادس والعشرون

اختبار وتصحيح البرنامج
Testing and Debugging

هناك أنواع متعددة من الأخطاء التي يمكن أن تظهر في البرنامج ، يمكن تصنيفها الى الأخطاء الهجائية (Misspelling) وأخطاء القواعد (Syntax Errors) والأخطاء المنطقية (Logical Errors) . والأخطاء الهجائية يمكن اكتشافها بسهولة عن طريق مراجعة قائمة الأوامر (List) واختبار هجاء كل كلمة للوصول الى هذه الأخطاء وتصحيحها . وكذلك فان أخطاء القواعد (Syntax Errors) تتطلب مراجعة كل أمر ومقارنته بالشكل الخاص به (Syntax) والذى تم وضعه بواسطة برنامج (+DBase III) أو برامج عائلة (DBase) الأخرى مثل (DBase IV) ، (+ FoxBase) ،(FoxPro) . وللرجوع الى الشكل (Syntax) الخاص بكل أمر يمكن استخدام شاشات المساعدة (Help Screens) التي توضح الشكل الخاص بكل أمر مع شرح كل الاختيارات الممكنة . أما الأخطاء المنطقية فانها تعتبر أعقد الأخطاء وأصعبها من حيث الاكتشاف أو التصحيح (Debugging) . حيث أنها أخطاء تتعلق بالتسلسل المنطقي للبرنامج وتسمى أحيانا أخطاء وقت التشغيل (Run Time Errors) ، لأنها لاتظهر الا عند التشغيل الفعلي للبرنامج . وبعض هذه الأخطاء لايؤدى الى توقف البرنامج ولكنه يؤدى الى الحصول على نتائج غير سليمة للبرنامج .

٣٦ - ١ خطوات الاختبار

كما سبق الايضاح ، فان تصميم البرنامج يعتمد على الطريقة التركيبية (Structured) .حيث يتم تقسيم البرنامج الى برامج فرعية (Modules) كل منها يؤدى مهمة محددة (Task). وهذه الطريقة تسهل كتابة البرنامج الى درجة كبيرة ، حيث أن كتابة البرنامج الفرعي الذى يؤدى مهمة محددة أسهل كثيرا من كتابة برنامج كبير يؤدى وظائف متعددة . كما يمكن في البرامج الكبيرة توزيع البرنامج على مجموعة من مخططي البرامج بحيث يقوم كل منهم بكتابة برنامج فرعي محدد . وكما تسهل هذه الطريقة كتابة البرامج ، فانها أيضا تسهل اختبارها وتصحيحها ، حيث يتم اختبار كل برنامج فرعي وتصحيحه مستقلا عن البرامج الأخرى .

وهناك خطوات قياسية (Standard) لاختبار وتصحيح أى برنامج . وهذه الخطوات يمكن تلخيصها فيما يلى :

١ - يتم كتابة كل برنامج فرعى (Module) وتوثيقه ، ثم اختباره بمجرد الانتهاء من كتابته ، ثم الانتقال الى البرنامج الفرعى التالى وكتابته ثم اختباره وتصحيحه وهكذا .
٢ - عند الانتهاء من مجموعة من البرامج التي تكون برنامجا مركبا (Composite) يتم ربطها معا وتشغيلها واختبار هذا البرنامج الجديد .
٣ - يتم تجميع البرامج المركبة فى برامج أكبر عن طريق اضافة البرامج الفرعية الجديدة (Modules) التى يتم اختبارها منفصلة .
٤ - عندما يتم تجميع البرنامج الكبير يتم اختباره أيضا بنفس الطريقة .

٥ - يتـــم اختبار البرنامج بواسطة أشخاص آخرين غير مشتركين في كتابة البرامج الفرعية . وهذا الاختبار يسمي اختبار ألفا (Alpha Testing) . وهو يتم عادة قبل البدء الفعلي في استخدام البرنامج . ويتم خلال هذا الاختبار تصحيح أى أخطاء تظهر في البرنامج .
٦ - يتم توزيع البرنامج على مجموعة محددة من المستخدمين (Users) ويقوم هؤلاء المستخدمون باختباره .وهذا الاختبار يسمى اختبار بيتا (Bita Testing).
٧ - بعد ذلك يصبح البرنامج جاهزا للاستخدام بواسطة مستخدمين آخرين . ولكن هذا لاينهى مرحلة الاختبار والتصحيح (Debugging) لأن هذه المرحلة تظل مستمرة ربما لعدة شهور أو عدة سنوات في بعض البرامج الكبيرة .

٢٦ - ٢ أوامر التصحيح (Debugging Commands)

يوفر البرنامج مجموعة من الأوامر التي تساعد مخطط البرامج على اكتشاف الأخطاء وتصحيحها . ويتوقف اختيار أى أمر من هذه الأوامر على طبيعة الخطأ المتوقع والطريقة المطلوبة لاكتشاف هذا الخطأ وتصحيحه . حيث أن بعض هذه الأوامر يؤدى الى تعليق تنفيذ أوامر البرنامج (Suspend) ، وبعضها يؤدي الى تنفيذ البرنامج خطوة خطوة حتى يتم اكتشاف الخطوة التي تسبب حدوث الخطأ .

٢٦ - ٣ تعليق تنفيذ البرنامج (Suspend)

عندما يحدث أى خطأ أثناء تشغيل البرنامج ، فان البرنامج يتوقف وتظهر الرسالة التالية :

Cancel , Ignore , Suspend (C, I, S)

واختيار الحرف (C) أى (CANCEL) يؤدى الى انهاء البرنامج والعودة الى مشيرة النقطة (Dot Prompt) .

واختيار الحرف (I) أي (IGNORE) يؤدى الى اكمال تنفيذ البرنامج وتخطى النقطة التي ظهر عندها الخطأ .

أما اختيار الحرف (S) أى (SUSPEND) فانه يؤدي الى توقف البرنامج مؤقتا والعودة الى مشيرة النقطة (Dot Prompt) . ومن هذا الوضع يمكن بحث الأخطاء المحتملة وتصحيحها من خلال مشيرة النقطة . وذلك عن طريق العودة الى قائمة البرنامج وتصحيح الأمر المتوقع حدوث الخطأ منه . وخلال هذه العملية يكون البرنامج معلقا ، أى أن تشغيله متوقف حتى يقوم مخطط البرامج بكتابة الأمر (RESUME) والضغط على مفتاح الادخال . حيث يستمر تنفيذ البرنامج مرة ثانية .

٣٦ - ٤ استخدام مخزن التاريخ (History)

ويقصد بمخزن التاريخ هنا مخزن ذاكرة مؤقت (Buffer) يتم فيه تخزين آخر عشرين أمرا تم ادخالها من خلال مشيرة النقطة . وفى أى وقت يراد رؤية الأوامر التى سبق ادخالها ، يستخدم مفتاح السهم لأعلى (↑) حيث أن كل ضغطة عليه تظهر الأمر السابق مباشرة ، وذلك بحد أقصى عشرين أمرا . ويتيح ذلك أيضا تنفيذ بعض الأوامر التى سبق ادخالها دون الحاجة الى كتابتها من جديد . حيث يكفى فى هذه الحالة الضغط على مفتاح (↑) عدة مرات حتى يظهر الأمر المطلوب ثم الضغط على مفتاح الادخال .

ويمكن زيادة الأوامر التى يمكن تخزينها فى هذا المخزن (History) عن طريق كتابة الأمر (SET HISTORY TO) ثم كتابة عدد الأوامر المطلوب تخزينها . فمثلا الأمر التالى :

```
SET HISTORY TO 30
```

يسمح بتخزين ٣٠ أمرا فى مخزن التاريخ . وهذا يؤدى الى الاحتفاظ بآخر ٣٠ أمرا تم ادخالها . ويمكن عرض هذه الأوامر كما سبق الايضاح باستخدام السهم (↑) وباستخدام الأمر (DISPLAY HISTORY) أو الأمر (LIST HISTORY) .

هذا ما يحدث بالنسبة للأوامر التى يتم ادخالها من خلال مشيرة النقطة (Dot Prompt) . أما أوامر البرنامج فانها فى الوضع المبدئى للبرنامج (Default) لايتم تخزينها فى مخزن التاريخ (History) . ولكن يمكن لمخطط البرامج تغيير هذا الوضع المبدئى عن طريق كتابة الأمر (SET DOHISTORY ON) ، وذلك قبل تشغيل البرنامج . وعند توقف البرنامج نتيجة وجود أى خطأ (Error) ، يمكن لمخطط البرامج عرض آخر عشرين أمرا تم ادخالها ، وذلك بكتابة الأمر التالى :

```
DISPLAY HISTORY
```

أو الأمر التالى :

```
LIST HISTORY
```

كما يمكن زيادة عدد هذه الأوامر باستخدام الأمر (SET HISTORY TO) كما سبق الايضاح .

ويجب ملاحظة أن استخدام الأمر (SET DOHISTORY ON) سوف يؤثر على كفاءة وسرعة تنفيذ البرنامج . لذلك يجب التأكد من اعادة هذا الأمر بعد اختبار البرنامج الى الوضع المبدئى (Default) كالآتى :

```
SET DOHISTORY OFF
```

٣٦ - ٥ مراقبة تنفيذ البرنامج

يتيح برنامج (+DBase III) لمخطط البرامج استخدام بعض الأوامر التي تساعده على مراقبة خطوات تنفيذ البرنامج . وهذا يساعده على اكتشاف مكان الخطأ وأحيانا التعرف على سبب هذا الخطأ.ويتم ذلك باستخدام الأمر (SET TALK ON) والأمر (SET ECHO ON) والأمر (SET DEBUG ON) .

٣٦ - ٦ الأمر (SET TALK ON)

الوضع المبدئي لهذا الأمر يكون (ON) ، وهذا يعني ظهور رسائل توضح تنفيذ كل أمر . ولكن عند كتابة البرنامج يتم تغيير هذا الوضع المبدئي عن طريق كتابة الأمر (SET TALK OFF). وذلك حتى لاتظهر هذه الرسائل للمستخدم عند تنفيذ البرنامج . ولذلك فعند تصحيح البرنامج (Debugging) يفضل اعادة الوضع المبدئي مرة ثانية حتى تظهر هذه الرسائل . وذلك لأنها تفيد عند حدوث خطأ في نقطة معينة من البرنامج حيث توضح هذه الرسائل أحيانا سبب الخطأ .

٣٦ - ٧ الأمر (SET ECHO ON)

وهذا الأمر يعتبر صورة مكبرة من الأمر (SET TALK ON) حيث أنه يؤدى الى ظهور خطوات تنفيذ كل أمر داخل الحاسب . وهذا يؤدى الى ظهور كل شيء ينفذه البرنامج على الشاشة أثناء تشغيله . فعند توقف البرنامج نتيجة خطأ معين (Error) يكون من السهل الوصول الى سبب هذا الخطأ .

٣٦ - ٨ الأمر (SET STEP ON)

يستخدم هذا الأمر لعرض خطوات تنفيذ البرنامج مثل الأمر (SET ECHO ON) تماما . ولكنه يختلف عنه في أنه يعرض هذه الخطوات خطوة خطوة . ويتوقف التنفيذ بعد كل خطوة حتى يضغط مخطط البرامج على أي مفتاح للانتقال الى الخطوة التالية . وهذه الطريقة كما هو واضح تؤدى الى ابطاء تنفيذ البرنامج بدرجة كبيرة . ولكنها تفيد في اكتشاف الأخطاء الدقيقة جدا التي يصعب اكتشافها باستخدام الوسائل السابقة .

٣٦ - ٩ الأمر (SET DEBUG ON)

يستخدم هذا الأمر أيضا في اكتشاف الأخطاء الدقيقة التي يصعب اكتشافها بالطرق السابقة . وهو يؤدى عند كتابته من مشيرة النقطة (Dot Prompt) الى

توجيه مخرجات الأمر (SET ECHO ON) أو مخرجات الأمر (SET STEP ON) الى الطابعة بدلا من عرضها على الشاشة . وهذا يؤدي الى الحصول على نسخة مطبوعة (Hard Copy) من أوامر البرنامج متضمنة خطوات تنفيذ كل أمر كما تحدث داخل الحاسب .

٣٦ - ١٠ عرض محتويات الذاكرة (Display Memory)

عند ظهور رسالة خطأ مثل (Variable Not Found) أو "Data Type Mismatch" فربما يكون سبب هذا الخطأ عدم انشاء متغير ذاكرة أو استخدام نوعين من البيانات في سطر واحد (بيانات حرفية مع بيانات عددية مثلا) . في هذه الحالة يمكن تعليق البرنامج (Suspend) ثم عرض محتويات الذاكرة باستخدام الأمر (DISPLAY MEMORY) . حيث يتم في هذه الحالة عرض جميع المتغيرات الموجودة في الذاكرة في هذه اللحظة مع تحديد حالة هذه المتغيرات .

ويمكن ايقاف البرنامج في أي وقت بالضغط على مفتاح الهروب (ESC) حيث تظهر الرسالة التالية :

```
Cancel , Ignore, Suspend (C,I,S)
```

ثم يتم اختيار (S) لتعليق البرنامج وعرض محتويات الذاكرة كما سبق الايضاح .

كما يمكن كتابة الأمر (SUSPEND) داخل البرنامج في المنطقة المشكوك في وجود خطأ فيها . وعند توقف البرنامج يتم كتابة الأمر (DISPLAY MEMORY) الذي يؤدي الى عرض محتويات الذاكرة باستخدام الأمر (DISPLAY MEMORY) كما سبق الايضاح .

٣٦ - ١١ عرض الحالة (Display Status)

هذا الأمر يؤدي الى عرض حالة البيئة (Environment) الخاصة بالبرنامج . وهذا يشمل عرض أسماء ملفات قواعد البيانات المفتوحة وملفات الفهرس الخاصة بها ، بالاضافة الى البيانات المبدئية لبرنامج (DBase III+) أو برامج عائلة (DBase) الأخرى مثل (DBase IV) ، (FoxBase +) ، (FoxPro) حسب البرنامج المستخدم .

٣٦ - ١٢ عرض تركيب ملف قاعدة البيانات

في بعض الأحيان يكون هناك أخطاء ناتجة عن اختيار حقول بيانات غير مطابقة

للمتغيرات المستخدمة في البرنامج . ويمكن اكتشاف هذه الأخطاء عن طريق عرض تركيب الملف بواسطة الأمر (DISPLAY STRUCTURE) . حيث يتم عرض حقول البيانات على الشاشة كالآتى مثلا:

Field	Field Name	Type	Width	Dec
1	Name	Character	30	
2	Address	Character	30	
3	Phone	Character	10	

كما يمكن طباعة هذا التركيب على الطابعة عن طريق كتابة الأمر التالى :

```
DISPLAY STRUCTURE TO PRINT
```

ملاحظة

ما سبق ذكره فى هذا الباب ينطبق أيضا على كل برامج عائلة (DBase) مثل (DBase IV) ، (FoxBase +) ، (FoxPro) .


الملاحق

ملحـــق (١)

أهم الأوامر والدوال المستخدمة

فى برنامج (DBase IV)

هذا الملحق يضم الأوامر والدوال الأضافية فقط . حيث أن باقى الأوامر والدوال مطابقة تماما لما سبق عرضه فى برنامج (+ DBase III) . مع ملاحظة أن الكتاب الثانى يحتوى على الشرح التفصيلى لجميع الأوامر والدوال .

أولا : الأوامـــر (Commands)

١ - الأمر (???)
وهو يؤدى الى ارسال المخرجات مباشرة الى الطابعة .

٢ - الأمر (@... SAY... GET)
وهو نفس الأمر الخاص ببرنامج (+DBase III) مع اضافة امكانيات أخرى له مثل ادخال شروط (Conditions) لظهور بيانات الـ (Gets) وظهور رسائل الأخطاء والتحكم فى ألوان الأعمدة الضوئية . ولمزيد من الايضاح ارجع الى الباب الخاص بالأوامر فى لكتاب الثانى.

٣ - الأمـر (@... FILL)
وهو يؤدى الى رسم مستطيل مظلل بلون محدد

٤ - الأمر (@... TO)
وهو نفس الأمر المستخدم فى برنامج(+DBase III) لرسم مستطيل على الشاشة مع بعض الاضافات للتحكم فى لون الخطوط .ارجع الى الباب الخاص بالأوامر فى الكتاب الثانى.

٥ - الأمر (ACTIVATE MENU)
ويستخدم لتشغيل عمود قوائم (Menu Bar) سبق تصميمه .

٦ - الأمر (ACTIVATE POPUP)
ويستخدم لتشغيل عمود قوائم (Menu Bar) مع القوائم المنبثقة منه (POPUP).

٧ - الأمر (ACTIVATE SCREEN)
وهو يؤدى الى العودة الى الشاشة الكاملة بدلا من النافذة المفتوحة (Active Window).

٨ - الأمر (ACTIVATE WINDOW)
ويستخدم فى تشغيل نافذة معينة بدلا من الشاشة الكاملة .

٩ - الأمر (APPEND FROM ARRAY)
ويستخدم فى اضافة سجلات الى ملف قاعدة البيانات من مصفوفة بيانات (Array).

١٠ - الأمر (APPEND MEMO)
ويستخدم في نقل بيانات من ملف نصي (Text File) الى حقل الملاحظات (Memo Field).

١١ - الأمر (ASSIST)
وهو يؤدي الى تشغيل مركز التحكم (Control Center) الخاص ببرنامج (DBase IV) .

١٢ - الأمر (BEGIN TRANSACTION)
ويستخدم فى بدء تعديل بيانات ملف قاعدة البيانات المفتوح بناء على بيانات ملفات الحركة .

١٣ - الأمر (CALCULATE)
ويستخدم فى عمل حسابات مالية واحصائية للبيانات العددية .

١٤ - الأمر (COMPILE)
وهو يستخدم فى تحويل البرنامج المكتوب بكود المصدر (Source Code) الى برنامج منفذ (Executable Program) .

١٥ - الأمر (COPY TO ARRAY)
ويستخدم فى ملء مصفوفة معينة بالبيانات الموجودة فى ملف قاعدة البيانات المفتوح .

١٦ - الأمر (CREATE APPLICATION)
ويستخدم لتشغيل مولد التطبيقات (Application Generator) الخاص ببرنامج (DBase IV) . وهو الذى يساعد على انشاء البرامج التطبيقية بواسطة برنامج (DBase IV) .

١٧ - الأمر (DEACTIVATE MENU)
ويستخدم فى مسح عمود القوائم الموجود على الشاشة والاحتفاظ به فى الذاكرة حتى يتم استخدامه مع الجمل الاختيارية (ON SELECTION)

١٨ - الأمر (DEACTIVATE POPUP)
ويستخدم فى مسح عمود القوائم والقوائم المتفرعة منه مع الاحتفاظ بها فى الذاكرة .

١٩ - الأمر (DEACTIVATE WINDOW)
ويستخدم فى مسح نوافذ معينة من الشاشة مع الاحتفاظ بها فى الذاكرة .

٢٠ - الأمر (DEBUG)
ويستخدم لتشغيل برنامج التصحيح (Debugger) الخاص ببرنامج (DBase IV).

٢١ - الأمر (DECLARE)
ويستخدم لانشاء مصفوفة أحادية أو ثنائية الأبعاد .

٢٢ - الأمر (DEFINE BAR)
ويستخدم هذا الأمر لتعريف أحد الاختيارات في عمود القوائم (Menu Bar)

٢٣ - الأمر (DEFINE BOX)
ويستخدم هذا الأمر فى انشاء مستطيل حول البيانات

٢٤ - الأمر (DEFINE MENU)
ويستخدم هذا الأمر فى انشاء القوائم (Menus) .

٢٥ - الأمر (DEFINE PAD)
ويستخدم فى تعريف أحد الاختيارات فى القائمة الفرعية (Popup Menu)

٢٦ - الأمر (DEFINE POPUP)
ويستخدم فى تعريف النافذة الفرعية (Popup Window) .

٢٧ - الأمر (DEFINE WINDOW)
ويستخدم فى تعريف نافذة معينة متضمنا الحدود والألوان .

٢٨ - الأمر (LIST/DISPLAY FILES)
ويستخدم لعرض معلومات عن الفهرس الحالى . وهو مثل الأمر (DIR) فى نظام التشغيل (DOS) .

٢٩ - الأمر (LIST/DISPLAY USERS)
ويستخدم لعرض حالة محطات العمل (Work Stations) التى تعمل مع البرنامج من خلال شبكة حاسبات (Network) .

٣٠ - الأمر (LOGOUT)
ويستخدم لفصل احدى المحطات (Work Stations) المتصلة من خلال شبكة (Network).

٣١ - الأمر (MOVE WINDOW)
ويستخدم لتحريك نافذة معينة على الشاشة .

٣٢ - الأمر (ON PAD)
ويستخدم لربط قائمة فرعية معينة بعمود اختيارات معين .

٣٣ - الأمر (ON PAGE)
ويستخدم للتحكم فى ناقل الصفحة (Page Break) . وكذلك فى العناوين العلوية (Headers) والعناوين السفلية (Footers) أثناء طباعة التقارير .

٣٤ - الأمر (ON READERROR)
ويستخدم في التعامل مع الأخطاء أثناء تشغيل البرنامج .

٣٥ - الأمر (ON SELECTION PAD)
ويستخدم في تنفيذ عمل معين عند اختيار أحد الاختيارات في احدى القوائم الفرعية .

٣٦ - الأمر (ON SELECTION POPUP)
ويستخدم في تنفيذ أمر معين عند اختيار أحد القوائم الفرعية (POPUP).

٣٧ - الأمر (PLAY MACRO)
ويستخدم في تنفيذ مجموعة من الأوامر التي سبق تخزينها في مجموعة من المفاتيح (Macro) .

٣٨ - الأمر (PRINTJOB)
ويستخدم لتنفيذ بعض الأوامر أثناء تنفيذ عملية الطباعة .

٣٩ - الأمر (PROTECT)
ويستخدم في تأمين بعض البيانات وحمايتها من الاستخدام الا بواسطة المختصين .

٤٠ - الأمر (RELEASE MODULE)
ويستخدم في مسح أي برنامج فرعي أو نافذة أو قائمة أو برنامج مكتوب بلغة التجميع من الذاكرة المؤقتة .

٤١ - الأمر (RESTORE MACRO)
ويستخدم لتحميل برنامج حروف مبرمجة (Macro) في الذاكرة المؤقتة .

٤٢ - الأمر (RESTORE WINDOW)
ويستخدم لتحميل نافذة من القرص الى الذاكرة المؤقتة .

٤٣ - الأمر (ROLLBACK)
ويستخدم لاعادة ملف قاعدة البيانات الى حالته الأولى قبل ادخال حركة معينة (Transaction).

٤٤ - الأمر (SAVE MACROS)
ويستخدم لتخزين برنامج المفاتيح المبرمجة (Macro) في ملف معين .

٤٥ - الأمر (SAVE WINDOW)
ويستخدم لتخزين نافذة معينة في ملف .

٤٦ - الأمر (SCAN)
ويستخدم لتنفيذ بعض الأوامر على مجموعة من السجلات .

٤٧ - الأمر (SHOW MENU)
ويستخدم لعرض قائمة على الشاشة دون تشغيلها .

٤٨ - الأمر (SHOW POPUP)
ويستخدم لعرض قائمة فرعية على الشاشة دون تشغيلها .

٤٩ - الأمر (UNLOCK)
ويستخدم لفتح الملف أو السجل حتى يصبح قابلا للتشغيل بواسطة المستخدم .

ثانيا : أوامر التجهيز (SET)

١ - الأمر (SET)
ويستخدم فى عرض قوائم الضبط

٢ - الأمر (SET AUTOSAVE)
ويسستخدم فى تخزين السجلات من المخزن المؤقت (Buffer) الى القرص .

٣ - الأمر (SET BELL TO)
ويستخدم فى التحكم فى الصوت الذى يحدث عند تشغيل الجرس .

٤ - الأمر (SET BORDER TO)
ويستخدم فى التحكم فى حدود النوافذ والقوائم الفرعية .

٥ - الأمر (SET CLOCK)
ويستخدم فى عرض ساعة النظام فى الركن العلوى الأيمن من الشاشة .

٦ - الأمر (SET CLOCK TO)
ويستخدم فى التحكم فى مكان عرض ساعة النظام .

٧ - الأمر (SET COLOR OF)
ويستخدم فى تحديد ألوان أشياء محددة على الشاشة .

٨ - الأمر (SET CURRENCY)
ويستخدم لعرض علامة العملة على اليمين أو اليسار حسب الحاجة .

٩ - الأمر (SET DATE)
ويستخدم فى تحديد شكل التاريخ للدول المختلفة .

١٠ - الأمر (SET DEVELOPMENT ON/OFF)
ويستخدم عند تطوير البرنامج وعمل بعض التعديلات . فعندما يكون كود المصدر (Source Code) قد تم تعديله بتاريخ يلى تاريخ كود الهدف (Object Code) يقوم برنامج (DBase IV) بإعادة ترجمة البرنامج (Compiling) .

١١ - الأمر (SET DISPLAY TO)
ويستخدم لضبط البرنامج على حالة الشاشة (Display Mode) اذا كانت (Mono) أو (EGA) أو (VGA) .

١٢ - الأمر (SET ENCRYPTION)
ويستخدم فى تشفير أو فك شفرة كود البرنامج الذى سبق حمايته بواسطة الأمر (PROTECT) .

١٣ - الأمر (SET HOURS TO)
ويستخدم في عـــرض الساعة وتحديد عدد الساعات بها اذا كان ١٢ أو ٢٤ ساعة .

١٤ - الأمر (SET LOCK)
ويستخدم لتشغيل أو عدم تشغيل قفل السجل . وهذا يعني التحكم في قدرة المستخدم على التعامل مع بيانات هذا السجل أو عدم قدرته على ذلك .

١٥ - الأمر (SET MARK TO)
ويستخدم لتحديد الحـرف الذى يفصل أرقام التاريخ . والحرف المبدئي هو الحرف (/) .

١٦ - الأمر (SET PAUSE)
ويستخدم في ايقاف الشاشة أو عدم ايقافها عند عرض بيانات تـزيـد عـن طول الشاشة .

١٧ - الأمـر (SET POINT TO)
ويستخدم لتحديد الحرف الذى يمثل العلامة العشرية . والحرف المبـدئي هـو النقطة (.) .

١٨ - الأمر (SET PRECISION)
ويستخدم لتحديد عدد الكسور العشرية التي تظهر في الأعداد . والعدد المبدئي هو (١٦) والمدى من ١٠ الى ٢٠ .

١٩ - الأمر (SET PRINTER)
ويستخدم في توجيه المخرجات الى الطابعة . أو عدم توجيهها .

٢٠ - الأمر (SET PRINTER TO)
ويستخدم في توجيه المخـرجات الى طابعـــة محـــددة . والطابعة المـبدئية هي (PRN) .

٢١ - الأمر (SET PRINTER TO FILE)
ويستخدم لتوجيه المخرجات الى ملف بدلا من الطابعة .

٢٢ - الأمر (SET WINDOW OF MEMO)
ويستخدم في تحديد النافذة التـــي يتـم عــن طريقها تعديل حقول الملاحظات (Memo Fields).

ثالثا : الدوال

١ - الدالة (ACCESS)
وهي دالة يتم عن طريقها تحديد مستويات التعامل مع البيانات للمستخدمين حسب درجة السرية الخاصة بكل منهم .

٢ - الدالة (ACOS)
وهي دالة تحسب مقدار الزاوية من جيب تمامها .

٣ - الدالة (ALIAS)
وهي دالة تحدد الاسم المرادف لمنطقة عمل محددة .

٤ - الدالة (ASIN)
وهي دالة تحسب مقدار الزاوية من جيبها . .

٥ - الدالة (ATAN)
وهي دالة تحسب مقدار الزاوية من ظلها .

٦ - الدالة (ATN2)
وهي دالة تحسب مقدار الزاوية من جيبها وجيب تمامها .

٧ - الدالة (BAR)
وهي دالة تعطي رقم آخر عمود قوائم (Menu Bar) تم اختياره .

٨ - الدالة (CEILING)
وهي دالة تحدد أصغر قيمة تزيد عن أو تساوي القيمة المحددة في الدالة .

٩ - الدالة (COMPLETED)
وهي دالة تحدد ما اذا كانت عملية ادخال الحركة (Transaction) قد تمت أم لا .

١٠ - الدالة (COS)
وهي دالة تحسب جيب تمام الزاوية .

١١ - الدالة (DIFFERENCE)
وهي تعطي الفرق بين سلسلتين حرفيتين .

١٢ - الدالة (DTOR)
وهي دالة تقوم بتحويل الزاوية من الدرجات (Degrees) الى الوحدات القطرية (Radians).

١٣ - الدالة (FILE)
وهى دالة تتحقق من وجود ملف معين .

١٤ - الدالة (FKLABEL)
وهى تحدد اسم مفتاح وظيفة معين (Function Key) من رقمه .

١٥ - الدالة (FKMAX)
وهى دالة تحدد أكبر عدد من مفاتيح الوظائف يمكن برمجته .

١٦ - الدالة (FLOCK)
وهى تستخدم فى اغلاق ملف قاعدة البيانات حتى لا يستطيع أى شخص التعامل معه .

١٧ - الدالة (FLOOR)
وهى دالة تحدد أكبر عدد أصغر من أو يساوى القيمة المحددة فى الدالة .

١٨ - الدالة (FV)
وهى دالة تعطى القيمة المستقبلية (Future Value) لاستثمار معين لفترة محددة وبمعدل ثابت .

١٩ - الدالة (LASTKEY)
وهى دالة تعطى كود الآسكى (ASCII Code) لآخر مفتاح تم الضغط عليه للخروج من شاشة معينة .

٢٠ - الدالة (LIKE)
وهى دالة تقوم بمقارنة سلسلتين حرفيتين . .

٢١ - الدالة (LINNO)
وهى تحدد رقم السطر الذى سوف يتم تنفيذه من البرنامج .

٢٢ - الدالة (LOCK)
وتستخدم فى اغلاق سجلات معينة من ملف قاعدة البيانات خلال شبكة الحاسبات (Network) حتى لا تستخدم بواسطة أى مستخدم .

٢٣ - الدالة (MENU)
وهى دالة تحدد اسم القائمة الفرعية المفتوحة .

٢٤ - الدالة (MESSAGE)
وهى دالة تحدد رسالة الخطأ الخاصة بآخر خطأ ظهر فى البرنامج . .

٢٥ - الدالة (MLINE)
وهى دالة تحدد سطرا معينا فى حقل الملاحظات .

٢٦ - الدالة (NETWORK)
وهى دالة تحدد ما اذا كان البرنامج يعمل على الشبكة (Network) أم لا .

٢٧ - الدالة (PAD)
وهى تحدد أسم القائمة الجارى العمل عليها .

٢٨ - الدالة (PAYMENT)
وهى دالة تحدد الأقساط المدفوعة لسداد قرض معين فى فترة معينة وبفائدة معينة .

٢٩ - الدالة (PI)
وهى دالة تعطى النسبة التقريبية (ط) .

٣٠ - الدالة (POPUP)
وهى دالة تعطى اسم القائمة الفرعية المفتوحة .

٣١ - الدالة (PRINTSTATUS)
وهى دالة تعطى حالة الطابعة الموصلة .

٣٢ - الدالة (PROGRAM)
وهى دالة تعطى اسم البرنامج الجارى تنفيذه عندما يظهر خطأ معين . وذلك لتحديد البرنامج الفرعى المحتوى على الخطأ .

٣٣ - الدالة (PV)
وهى تعطى القيمة الحالية لقرض يتم تسديده على أقساط فى فترات زمنية محددة .

٣٤ - الدالة (RAND)
وهى دالة تقوم بتوليد أعداد عشوائية .

٣٥ - الدالة (RTOD)
وهى تستخدم لتحويل الزوايا من التقدير الدائرى (Radian) الى درجات .

ملحق (٢)

أهم الأوامر والدوال المستخدمة

فى برنامج (FoxBase+)

هذا الملحق يضم الاوامر والدوال الاضافية فقط . حيث ان باقي الاوامر و الدوال مطابقة تماما لما سبق عرضه في برامج (+DBaseIII) ، (DBaseIV) . مع ملاحظة ان الكتاب الثاني يحتوى على الشرح التفصيلي لجميع الاوامر و الدوال .

أولا : الأوامــر

١ - الأمر (CLEAR PROGRAM)
ويستخدم في مسح برنامج مكتوب بلغة التجميع (Assembly) .

٢ - الأمر (CLEAR TYPEAHEAD)
ويستخدم لمسح مخزن الذاكرة المؤقت (Buffer) .

٣ - الأمر (DELETE FILE)
ويستخدم في مسح ملف معين من القرص .

٤ - الأمر (DIMENSION)
ويستخدم في انشاء مصفوفة متغيرات أحادية أو ثنائية .

٥ - الأمر (DIR)
ويستخدم في عرض فهرس الملفات .

٦ - الأمر (FLUSH)
ويستخدم في نقل محتويات الذاكرة الى القرص .

٧ - الأمر (GATHER FROM)
ويستخدم في تخزين بيانات المصفوفة في متغيرات ذاكرة .

٨ - الأمر (MENU)
ويستخدم لتشغيل عمود أختيارات (Menu Bar) .

٩ - الأمر (ON KEY = <expN>)
ويستخدم في تنفيذ مجموعة من الأوامر عند ضغط المستخدم على مفتاح محدد .

١٠ - الأمر (RELEASE MODULE)
ويستخدم في مسح برنامج سبق تحميله في الذاكرة .

١١ - الأمر (RESTORE SCREEN)
ويستخدم لتحميل شاشة من الذاكرة أو من متغيرات ذاكرة معينة .

١٢ – الأمر (SAVE SCREEN)
ويستخدم لتخزين الشاشة فى متغير ذاكرة محدد .

١٣ – الأمر (SCATTER)
ويستخدم فى نقل محتويات متغيرات ذاكرة الى مصفوفة متغيرات .

١٤ – الأمر (UNLOCK)
ويستخدم فى مسح كل أقفال الملفات والسجلات حتى يمكن التعامل معها بواسطة أى مستخدم .

ثانيا : الدوال

١ - الدالة (ALIAS)
وهى تعطى الاسم المرادف لمنطقة العمل المفتوحة .

٢ - الدالة (FCOUNT)
وهى تعطى عدد الحقول الموجودة فى ملف قاعدة البيانات المفتوح .

٣ - الدالة (FILE)
وهى تختبر وجود ملف معين على القرص .

٤ - الدالة (FKLABEL)
وهى تعطى اسم مفتاح الوظيفة (Function Key) المقابل لرقم معين .

٥ - الدالة (FKMAX)
وهى تعطى أكبر عدد يمكن برمجته من مفاتيح الوظائف .

٦ - الدالة (FLOCK)
وهى تستخدم فى اغلاق ملف قاعدة بيانات محدد حتى لايتم استخدامه بواسطة باقى المستخدمين .

٧ - الدالة (LOCK)
وهى تستخدم فى اغلاق السجل المفتوح حتى لايتم استخدامه بواسطة المستخدمين الآخرين .

٨ - الدالة (SYS(0))
وهى تعطى اسم الجهاز ورقمه فى شبكة الحاسبات .

٩ - الدالة (SYS(1))
وهى تعطى التاريخ بالشكل المستخدم فى برنامج (FoxBase) وبرنامج (DBase II) .

١٠ - الدالة (SYS(2))
وهى تعطى عدد الثوانى المنقضية بعد منتصف الليل وحتى الوقت الحالى . أى أنها تعطى الوقت الحالى .

١١ - الدالة (SYS(3))
وهى تعطى اسم ملف مؤقت يتم استخدامه بواسطة أحد المستخدمين المتصلين بالشبكة .

١٢ - الدالـة ((SYS(5))
وهى تعطى اسم وحدة الأقراص المبدئية (Default) .

١٣ - الدالـة ((SYS(6))
وهى تعطى الطابعة الحالية (Current Printer) .

١٤ - الدالـة ((SYS(7 , | W |))
وهى تعطى اسم ملف التشكيل المفتوح (Format File) فى منطقة العمل الممثلة بالرقم (W) .

١٥ - الدالـة ((SYS(9))
وهى تعطى رقم نسخة برنامج (+ FoxBase) .

١٦ - الدالـة ((SYS(10,d))
وهي تحول الرقم الممثل للتاريخ (d) الى تاريخ حرفي (Character Date) مطابـــق في الشكل للتاريخ المستخدم بواسطة برنـامـج (FoxBase) وبرنامج (DBase II) .

١٧ - الدالـة ((SYS(11,s))
وهى عكس الدالة السابقة . حيث تعطى الرقم الممثل للتاريخ الحرفى (s) .

١٨ - الدالـة ((SYS(12))
وهى تعطى الذاكرة المؤقتة المتاحة .

١٩ - الدالـة ((SYS(13))
وهى تعطى حالة الطابعة اذا كانت جاهزة أو غير جاهزة .

٢٠ - الدالـة ((SYS(14,n ,w))
وهي تعطى الملف الفهرسى رقم (n) في منطقة العمـل (W) مـع مـلاحـظة أن العدد (n) لايزيد عن (٧) والعدد (W) لايزيد عن (١٠) .

٢١ - الدالـة ((SYS(15,t,s))
وهى دالة تعطى السلسلة الحرفية (s) المقابلة للسلسلة الحرفية (t) حسب جدول التحويل الخاص بالمستخدم (Translation Table) .

٢٢ - الدالـة ((SYS(16,n))
وهى دالة تعطى اسم البرنامج الجارى تنفيذه . وهى تفيد عند متابعة خطأ معين ومحاولة التوصل الى البرنامج المتسبب فى هذاالخطأ . والعدد (n) يمثل المستوى الذي يقع عليه البرنامج . فالعدد (١) يمثل البرنامج الرئيسي (Master Program) واذا تم حذف العدد (n) فان الدالة تعطى البرنامج الحالى الجارى تنفيذه .واذا أريد متابعة جميع البرامج الفرعية يتم استخدام الأوامر التالية :

```
STORE 1 TO i
DO WHILE LEN(SYS(16,i)<>0)
     ? SYS(16,i)
     STORE i+1 TO i
ENDDO
```

٢٣ - الدالة ((SYS(17))
وهي تعطي نوع المعالج الدقيق المستخدم في الجهاز اذا كان (8086/8088) أو (80286) او (80386) .

٢٤ - الدالة ((SYS(18))
وهي تعطي اسم حقل قاعدة البيانات الذي يتم ادخاله عن طريق الأمر (@...GET) . وذلك عند ضغط المستخدم على المفتاح الذي تم تحديده بواسطة الأمر (ON KEY=t) حيث (t) يمثل اسم المفتاح .

٢٥ - الدالة ((SYS(100))
وهي تعطي حالة الشاشة (Console) اذا كانت (ON) أو (OFF) .

٢٦ - الدالة ((SYS(101))
وهي تعطي حالة الجهاز المستخدم اذا كان الشاشة أو الطابعة .

٢٧ - الدالة ((SYS(102))
وهي تعطي حالة الطابعة (Printer) اذا كانت (ON) أو (OFF) .

٢٨ - الدالة ((SYS(103))
وهي تعطي حالة الأمر (SET TALK) اذا كانت (ON) أو (OFF) .

٢٩ - الدالة (UPDATED)
وهي تعطي حالة البيانات اذا كان قد تم تحديثها أم لا .

٣٠ - الدالة (VERSION)
وهي تعطي رقم نسخة برنامج (+ FoxBase) المستخدم .

ملحق (٣)

أهم الأوامر و الدوال المستخدمة

فى برنامج (FoxPro)

هذا الملحق يضم الأوامر والدوال الأضافية فقط . حيث أن باقي الأوامر والدوال مطابقة تماما لما سبق عرضه في برامج (+ FoxBase) ، (DBase IV) ، (+ DBase III) . مع ملاحظة أن الكتاب الثاني يحتوى على الشرح التفصيلي لجميع الأوامر والدوال .

أولا : الأوامــر

١ - الأمــر (COPY MEMO)
ويستخدم لنسخ محتويات حقل الملاحظات في ملف

٢ - الأمر (FILER)
ويستخدم في تشغيل برنامج صيانة الملفات الخاص ببرنامج (FoxPro)

٣ - الأمر (FOR)
ويستخدم في انشاء حلقة تكرارية مع تشغيل عداد لحساب عدد مرات تنفيذ الحلقة .

٤ - الأمــر (FUNCTION)
وهو أمر يحدد بداية برنامج فرعي (Subroutine) .

٥ - الأمــر (MODIFY MEMO)
ويستخدم في تعديل محتويات حقل الملاحظات .

٦ - الأمــر (SET COLOR SET)
ويستخدم في تحميل مجموعة من الألوان سبق تحديدها .

٧ - الأمــر (SET CURSOR)
ويستخدم في التحكم في مؤشر الشاشة .

٨ - الأمــر (SET LOGERRORS)
ويستخدم في ارسال أخطاء البرنامج أثناء ترجمته الى ملف على القرص .

٩ - الأمر (SET MACKEY TO)
ويستخدم في تحديد مفتاح أو عدة مفاتيح لتشغيل برنامج الماكرو .

١٠ - الأمــر (SET MOUSE)
ويستخدم في تشغيل الفأرة (Mouse) .

١١ - الأمــر (SET MOUSE TO)
ويستخدم في التحكم في الفأرة (Mouse).

١٢ - الأمـــر (SET RELATION OFF INTO)
ويستخدم فى الغاء علاقة بين ملفين مفتوحين .

١٣ - الأمـــر (SET SHADOWS)
ويستخدم فى التحكم فى ظهور ظل النافذة أو عم ظهوره .

ثانيا : الدوال

١ - الدالة (ALLTRIM)
وتستخدم فى حذف المسافات الخالية من أول السلسلة الحرفية وآخرها .

٢ - الدالة (ATC)
وتستخدم فى البحث عن سلسلة حرفية داخل سلسلة حرفية أخرى بصرف النظر عن حالة الحروف اذا كانت صغيرة أو كبيرة . وهى تعطى عددا يمثل مكان هذه السلسلة .

٣ - الدالة (ATCLINE)
وهى تبحث عن سلسلة حرفية معينة داخل سلسلة حرفية أخرى بصرف النظر عن حالة الحروف اذا كانت صغيرة أو كبيرة . وهى تعطى رقم السطر المحتوى على هذه السلسلة .

٤ - الدالة (ATN2)
وهى تعطى قيمة الزاوية بالتقدير الدائرى من جيب الزاوية وجيب تمامها .

٥ - الدالة (CAPSLOCK)
وهى تعطى حالة مفتاح الحروف الكبيرة (Caps) .

٦ - الدالة (CHRSAW)
وهى تحدد ما اذا كان أحد الحروف موجودا فى مخزن الذاكرة الخاص بلوحة المفاتيح .

٧ - الدالة (CHRTRAN)
وهى دالة تقوم بترجمة حرف أو حروف معينة باستخدام جداول ترجمة محددة (Translation Tables) .

٨ - الدالة (CURDIR)
وهى تعطى الفهرس الحالى (Current Directory) .

٩ - الدالة (DATE)
وهى تعطى تاريخ اليوم الحالى .

١٠ - الدالة (DMY)
وهى دالة تقوم بتحويل التاريخ الى الشكل (Day Month Year) .

١١ - الدالة (EMPTY)
وهى تحدد اذا كانت السلسلة خالية (Blank) أم لا .

١٢ - الدالـة (FCOUNT)
وهى تعطى عدد الحقول فى ملف قاعدة البيانات المفتوح .

١٣ - الدالـة (FGETS)
وهى تعطى حروفا من ملف حتى يتم الضغط على مفتاح الادخال .

١٤ - الدالـة (FILTER)
وهى تعطى الشروط المستخدمة فى المرشح (Filter) المستخدم .

١٥ - الدالـة (FOPEN)
وتستخدم فى فتح ملف .

١٦ - الدالـة (FPUTS)
وهى تقوم بكتابة سلسلة حرفية فى الملف المفتوح .

١٧ - الدالـة (FREAD)
وهى تقوم بقراءة عدد من الحروف من الملف المفتوح .

١٨ - الدالـة (FSIZE)
وهى تعطى حجم حقل معين بالحروف (Bytes) .

١٩ - الدالـة (FULLPATH)
وهى تعطى المسار الخاص بملف معين .

٢٠ - الدالـة (FWRITE)
وهى تكتب سلسلة حرفية معينة فى الملف المفتوح .

٢١ - الدالـة (HEADER)
وهـــى تعطى عدد الحروف الموجودة فى العنوان (Header) الخاص بأحد الملفات .

٢٢ - الدالـة (INLIST)
وهى تجدد ما اذا كانت سلسلة حرفيهمعينة موجودة ضمن مجموعة من السلاسل الحرفية .

٢٣ - الدالـة (ISDIGIT)
وهى تحدد ما اذا كان أول حرف من سلسلة حرفية رقما أم حرفا .

٢٤ - الدالـة (MCOL)
وهى تعطى مكان العمود الخاص بمؤشر نافذة معينة .

٢٥ - الدالـة (MDY)
وهى تحول التاريخ الى صورة (شهر يوم سنة)أى(Month, Day, Year).

٢٥ – الدالـة (MEMLINES)
وهى تعطى عدد السطور فى حقل الملاحظات .

٢٦ – الدالـة (MEMORY)
وهى تعطى الذاكرة المؤقتة المتاحة .

٢٧ – الدالـة (MESSAGE)
وهى تعطى رسالة الخطأ المستخدمة .

٢٨ – الدالـة (MROW)
وهى تعطى رقم العمود الخاص بمؤشر نافذة معينة .

٢٩ – الدالـة (OCCURS)
وهى تعطى عدد مرات وجود سلسلة حرفية معينة داخل سلسلة حرفية أخرى .

٣٠ – الدالـة (ORDER)
وهى تعطى اسم الفهرس الرئيسى فى منطقة عمل محددة .

٣١ – الدالـة (PARAMETERS)
وهى تعطى عدد المعاملات التى تم ادخالها الى آخر برنامج خطوات (Procedure) .

٣٢ – الدالـة (PROPER)
وهى تعطى الأسماء بشكلها المتعارف عليه . وهو أن يكون أول حرف كبيرا (Capital) وباقى الحروف صغيرة (Small) .

٣٣ – الدالـة (RATLINE)
وهى تبحث عن وجود سلسلة حرفية معينة داخل سلسلة حرفية أخرى وتحدد رقم السطر الموجودة به .

٣٤ – الدالـة (RELATION)
وهى تعطى العلاقة التى سبق تكوينها فى منطقة العمل المفتوحة .

٣٥ – الدالـة (RTOD)
وهى تحول الزاوية من التقدير الدائرى الى درجات .

٣٦ – الدالـة (SCOLS)
وهى تعطى عدد الأعمدة المتاحة على الشاشة .

٣٧ – الدالـة (SIGN)
وهى تحدد الاشارة الخاصة بالعدد سواء كانت سالبة أو موجبة .

٣٨ – الدالة (SROWS)
وهى تعطى عدد السطور المتاحة على الشاشة .

٣٩ - الدالـة ((SYS(23)
وهى تعطى حجم الذاكرة الممتدة (Extended) المستخدمة فى برنامج (FoxPro).

٤٠ - الدالـة ((SYS(24)
وهى تعطى حجم الذاكرة الممتدة (Extended) المتاح والذى سبق تحديده من خلال ملف المواصفات (CONFIG.SYS) .

٤١ - الدالة ((SYS(1001)
وهى تعطى حجم الذاكرة المتاحة لبرنامج (FoxPro) .

٤٢ - الدالـة ((SYS(2000,<expC>)
وهى تعطى اسم الملف المطابق للسلسلة الحرفية (<expC>) .

٤٣ - الدالـة ((SYS(2001,<expC>)
وهى تعطى الحالة الحالية (Current Status) للأمر (SET) الممثل بالسلسلة الحرفية (<expC>) .

٤٤ - الدالـة ((SYS(2002)
وهى تتحكم فى تشغيل أو عدم تشغيل مؤشر الشاشة (Cursor) .

٤٥ - الدالـة ((SYS(2003)
وهى تعطى اسم الفهرس الحالى المستخدم .

٤٦ - الدالـة ((SYS(2005)
وهى تعطى أسم ملف الموارد المستخدم (Resource File) .

٤٧ - الدالـة] (SYS(2006)
وهى تعطى نوع الشاشة المستخدمة والكارت المستخدم معها .

٤٨ - الدالـة ((SYS(2007,<expC>)
وهى تعطى نتيجة اختبار الجمع (Check Sum) للسلسلة الحرفية < expC> .

٤٩ - الدالـة ((SYS(2008)
وهى تحدد شكل مؤشر التصحيح المستخدم .

٥٠ - الدالـة ((SYS(2009)
وهى تحول بين وضع الاضافة (Insert) ووضع الكتابة الفوقية (Overwrite).

٥١ - الدالـة (WCOLS)
وهى تعطى عدد الأعمدة فى نافذة معينة .

٥٢ - الدالـة (WEXIST)
وهي تحدد ما اذا كانت نافذة معينة قد تم تعريفها أم لا .

٥٣ - الدالـة (WLCOL)
وهي تعطي رقم العمود الخاص بنافذة معينة .

٥٤ - الدالـة (WLROW)
وهي تعطي رقم الصف الخاص بنافذة معينة .

٥٥ - الدالة (WOUTOPUT)
وهي تحدد ما اذا كانت المخرجات قد تم توجيهها الى نافذة معينة أم لا .

٥٦ - الدالة (WROWS)
وهي تعطي عدد السطور في نافذة معينة .

ملحق (٤)

جدول مقارنة
نظم ادارة قواعد البيانات

جـــــــدول المقارنـــــــة

عنـــاصر المقارنـــة	Clipper	DBase III Plus	DBase IV	FoxBase Plus	FoxPro
PROGRAMMING ENVIRONMENT					
Creates .Exe files	*				
Tokenises or Pseudo-complies			*	*	*
Run-time version		*	*	*	*
Interactive environment		*	*	*	*
Integrated text editor		*	*	*	*
Debugger	*		*		*
Documentation generator				*	*
Disc-management DOS shell			*		
FORMS AND REPORTS					
Form generator		*	*	*	*
Report generator		*	*	*	*
Label generator		*	*	*	*
Query by example			*		
DATABASE STRUCTURE					
Date fields	*	*	*	*	*
International date formats	*	*	*	*	*
Time fields	*	*	*	*	*
Memo fields	*	*	*	*	*
Program accesses strings within memos	*		*		*
Stores and displays graphics within database					
USER-INTERFACE PROGRAMMING					
Full-screen record editing	*	*	*	*	*
Range checks during editing	*	*	*	*	*
Programmer-defined validation during editing	*		*	*	*
Timeouts during record editing	*				*
Switches between table and form durintg editing			*		*
Program traps hotkeys	*	*	*	*	*
Programmer-redefinable function keys	*	*	*	*	*
Tests for keystroke without pausing	*	*	*	*	*
Programmer-defined light-bar menus	*		*	*	*
Programmer-defined pull-down menus			*	*	*
Programmer-defined scrolling windows	*		*	*	*
Programmer-defined help	*				*
USER INPUT/OUTPUT					
Mouse support within program					*
Keyboard macros			*		*
Programmer-definable screen colours	*	*	*	*	*
Changes screen colour without overwriting text			*		*
Supports 45-line EGA and 50-line VGA			*		*
PROGRAM INTERFACING					
Arrays	*		*	*	*
Case statement	*	*	*	*	*
User-defined functions	*		*	*	*
User-defined commands					
PROGRAM INTERFACING					
Calls assembler routines	*	*	*	*	*
Runs DOS programs	*	*	*	*	*
DOS file I/O	*				*
Creates TSR programs					
Programmer can trap errors		*	*	*	*

نظم ادارة قواعد البيانات

(الجزء الأول)

رقم الإيداع ١٩٩١/١٩٧٠

مطابع المكتب المصرى الحديث
MODERN EGYPTIAN PRESS

نظم إدارة قواعد البيانات